StreamServe™ Persuasion™

StreamServe Persuasion SP4 Drivers

User Guide

Rev A

StreamServe Persuasion SP4 Drivers User Guide
Rev A
© 2001-2009 STREAMSERVE, INC.
ALL RIGHTS RESERVED
United States patent #7,127,520

No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of StreamServe, Inc. Information in this document is subject to change without notice. StreamServe Inc. assumes no responsibility or liability for any errors or inaccuracies that may appear in this book. All registered names, product names and trademarks of other companies mentioned in this documentation are used for identification purposes only and are acknowledged as property of the respective company. Companies, names and data used in examples in this document are fictitious unless otherwise noted.

StreamServe, Inc. offers no guarantees and assumes no responsibility or liability of any type with respect to third party products and services, including any liability resulting from incompatibility between the third party products and services and the products and services offered by StreamServe, Inc. By using StreamServe and the third party products mentioned in this document, you agree that you will not hold StreamServe, Inc. responsible or liable with respect to the third party products and services or seek to do so.

The trademarks, logos, and service marks in this document are the property of StreamServe, Inc. or other third parties. You are not permitted to use the marks without the prior written consent of StreamServe, Inc. or the third party that owns the marks.

Use of the StreamServe product with third party products not mentioned in this document is entirely at your own risk, also as regards the StreamServe products.

StreamServe Web Site
http://www.streamserve.com

Contents

AFP **7**
Configuring the AFP driver **9**
Job Begin AFP driver settings 10
Document Begin AFP driver settings 18
Page Begin AFP driver settings 19
Configuration file AFP driver settings 20
Modes for managing AFP resources **22**
Fonts and code pages **25**
Generating font resources 25
Code pages 26
Generating code page resources 27
Mapping fonts and code pages 27
Images and overlays **31**
Generating image resources 31
Generating overlay resources 32
Mapping images and overlays 33
Form definitions **36**
Variables and metadata **37**
Page identification parameters included in the generated AFP data stream file **38**
Optimizing performance **39**
Upgrading an AFP (StreamServe 3.x) driver **40**
HTML **41**
IJPDS **45**
Configuring the IJPDS driver **47**
Preparing the configuration file 47
Resolution **49**
Color management **50**
Using color lookup tables 50
Using specific color lookup tables for individual images 51
Using ICC profiles for RGB to CMYK conversion 52
Color separation **54**
Configuring the color a RIP should print 54
Printing color registration mark 56
Replacing rich black 56
Replacing CMY black 57
Configuring the total ink amount 58
Halftoning **59**
Printheads and RIPs **60**

Configuring printheads 60
Using a RIP with single printhead 61
Using RIPs with stitched printheads 62
Using stitched RIPs 63
Printing cue marks 64
Generating a RIP setup file 65
Duplex and n-up printing 66
Converting overlays to fixed files 68
Compressing bitmap 69
Allocating resource memory 70
Splitting large jobs 71
IJPDS driver quick reference 72
PCL 6 77
Output Connector Settings 77
Runtime Output Connector Settings 81
PDF 83
PDF driver settings 85
Encryption 90
PDF (PCL Convert) 91
PDF (PCL Convert) driver settings 92
PostScript 93
XPS 95
Zebra Label Printer 97
Upgrading existing Projects using the Zebra Label Printer driver 98
Driver files and printer resolution 99
Getting the label size right 100
Fonts 101
Mapping fonts to Zebra fonts 101
Font sizes 102
Downloading TrueType and Unicode fonts to the Zebra printer 103
Configuring the Zebra printer 104
Editing the driver file and adding the fonts to be downloaded to the printer 105
Printing white text on black background 106
Barcodes 107
Graphics 109
Enabling hexadecimal character values 110
Zebra Label Printer Settings 111
Specifying fonts 115
Adding font entries to driver files 117
Adding TTF font entries to driver files 117
True Type Collections (TTC) 118
Adding TTF font entries to PostScript, PDF, and raster driver files 118

Adding TTF font entries to PCL driver files 118
Adding Type 1 font entries to driver files 119
Adding character encodings to Type1 font entries 120
Embedding fonts and downloading soft fonts 121
Embedding fonts in PDF documents 121
Embedding fonts in output to PCL printers 121
Downloading soft fonts to PCL printers 122

RFID 125
Supported air protocols and printer models 127
RFID settings 128

AFP

The AFP driver is used to send output to an AFP printing environment. The AFP driver creates an AFP data stream file for each job produced by the StreamServer.

Figure 1 AFP data stream file structure (simplified)

AFP resources

Fonts, code pages, overlays, images, and form definitions, can be represented in different ways in an AFP data stream file.

- Wrapped into a resource group

  A resource group is similar to a zip file containing uniquely named resources. The resources are either generated by the StreamServer, or retrieved from an external storage, and then wrapped into the resource group.

- As a reference in a page definition

  The page definitions in the AFP data stream file (`Page` below `Page group` in *Figure 1*) include references to resources. When the Print Server reads the AFP data stream file and finds a reference, it searches for the resource in the resource group. If the Print Server finds the resource in the resource group, it uses that resource.

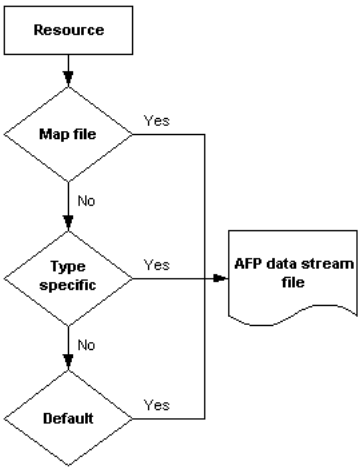
  If the resource is not included in the resource group, the Print Server searches for the resource in the local resource storage. If the Print Server finds the resource in the local storage, it uses that resource. If the Print Server cannot find the resource, it issues an error message.

- As a bitmap in a page definition (only images and overlays)

  Images and overlays can be merged as bitmaps directly on a page definition.

How the StreamServer determines the resource options to use

Resource names, resource storage location, and resource modes (include, embed, reference, etc.) can be specified in different ways. In the GUI settings for the AFP driver you can specify default options that apply to all resource types, and options that apply per resource type. You can also specify resource options in map files. The flow chart below illustrates how the StreamServer determines the options to use when adding a resource to the AFP data stream file.



Required knowledge

You must be familiar with basic AFP terminology.

Configuring the AFP driver

To enable an output connector to deliver AFP output, you must select **Output mode** > **Job**, and configure the AFP driver for the connector.

Platform driver settings

In the Platform connector configuration you must select the AFP driver. You cannot configure any driver settings at this stage.

Document Begin driver settings

In the Runtime connector configuration, at Document Begin, you can configure document related NOP comments and TLE indexes. See *Document Begin AFP driver settings* on page 18.

Job Begin driver settings

In the Runtime connector configuration, at Job Begin, you configure all other AFP driver settings. See *Job Begin AFP driver settings* on page 10.

Configuration file driver settings

In the driver configuration file, you can configure user defined halftone matrix and halftone transfer curve. This is only recommended for experts in this area. See *Configuration file AFP driver settings* on page 20.

Job Begin AFP driver settings

Settings	
Resolution	The resolution (dpi) of images, overlays, and fixed raster fonts in the AFP data stream output. This is ignored for resources using Reference mode. Which option to select depends on the capabilities of the printer.
Color	The color range to use for text, graphics, and images in the AFP data stream output. Which option to select depends on the capabilities of the printer. **None** – All colors are mapped to black and white. **Yes** – Text and graphics colors are mapped to a limited set of colors. Images are mapped to black and white. **Extended** – All colors are mapped to the RGB 0-255 color model. See also *.Technology (Image)* for information about image colors. **Grayscale** – All colors are mapped to 256 shades of gray. **Standard Limited for PTOCA** – All colors used for PTOCA objects are mapped to a limited OCA color set containing the following color values: • Blue • Red • Pink/Magenta • Green • Turquoise/Cyan • Yellow • Black • Brown • Device default • Color of medium **CMYK** – All colors are mapped to the CMYK color model.

Settings	
GOCA	The GOCA (Graphics Object Contents Architecture) options apply to vector graphics drawn in Storyteller, PageOUT or in the Overlay Editor. Which option to select depends on the capabilities of the printer. **None** – GOCA is not used to generate vector graphics. Only vertical and horizontal lines are presented in the output. **Yes** – Algorithms with simple GOCA objects are used to generate vector graphics. Enables printing of free lines, polygons, ovals, and round corners. **Extended** – The extended GOCA set is used to generate vector graphics without GOCA fillets. Enables printing of free lines, polygons, ovals, and round corners. **Full** – The extended GOCA set is used to generate vector graphics with GOCA fillets. Enables printing of free lines, polygons, ovals, and round corners. **Raster All** – All vector graphic objects on pages or overlays are rasterized. Enables printing of free lines, polygons, ovals and round corners as raster images. **Raster Patterns** – All vector graphic objects with patterns are rasterized. GOCA supports only a limited set of patterns. This option allows rasterizing of objects filled by unsupported patterns.

Settings	
.Color (GOCA)	**Default** – Set GOCA colors according to global color setting (**Color** option). **None** – All colors for GOCA objects are mapped to black & white. **Yes** – Text and graphics colors are mapped to a limited set of colors. Images are mapped to black and white. **Extended** – All colors for GOCA objects are mapped to the RGB 0-255 color model. See also *.Technology (Image)* for information about image colors. **Grayscale** – All colors for GOCA objects are mapped to 256 shades of gray. **Standard Limited** – All colors used for GOCA objects are mapped to a limited OCA color set containing the following color values: • Blue • Red • Pink/Magenta • Green • Turquoise/Cyan • Yellow • Black • Brown • Device default • Color of medium **CMYK** – All colors for GOCA objects are mapped to the CMYK color model.
.Rasterizing Threshold (GOCA)	The lower limit of when the GOCA object is rasterized. If the number of points (where there is one point per line of a polygon or shape and three points per Bezier curve) of the object is lower than this value for the incoming GOCA objects, it will not be rasterized.
.Line thickness factor (GOCA)	The line thickness factor applies to polygons only – not to lines and boxes. It affects the line width of polygons in the output. The line width is expressed as parts of an inch, and the actual line width in the output depends on the output device. In most cases you can keep the default line thickness factor (100), but you may have to change the value in order to tune the line width in the output. For example, if there are problems with the line widths in business graphics, you can modify the line thickness factor to solve the problem.

Settings	
.Line maximum width (GOCA)	The maximum width of lines to generate using the Line command. Lines wider than this are generated using the Area command. The GOCA commands Line and Area are used to generate lines and areas in the output. The Line command is device dependent, but the Area command is not.
.Pattern Set Limited (GOCA)	If you use fill functionality with less than 20% shading, you may see a dot pattern instead of a shading in the filled area. To avoid this you can limit the GOCA pattern set to use only the four first patterns.
Job comment	A comment that is included at the beginning of the job. Any string is accepted. The comment is translated into a NOP object in the AFP data stream. You can include metadata in the string (see *Assigning metadata values to comments and TLE indexes* on page 37).
End Job comment	A comment that is included at the end of the job. Any string is accepted. The comment is translated into a NOP object in the AFP data stream. You can include metadata in the string (see *Assigning metadata values to comments and TLE indexes* on page 37).
Before document group comment	A comment that is included before all document groups. Any string is accepted. The comment is translated into a NOP object in the AFP data stream. You can include metadata in the string (see *Assigning metadata values to comments and TLE indexes* on page 37).
Document group comment	A comment that is included before each document group. Any string is accepted. The comment is translated into a NOP object in the AFP data stream. You can include metadata in the string (see *Assigning metadata values to comments and TLE indexes* on page 37).
End document group comment	A comment that is included after all document groups. Any string is accepted. The comment is translated into a NOP object in the AFP data stream. You can include metadata in the string (see *Assigning metadata values to comments and TLE indexes* on page 37).
Resource	The default mode for how to handle resources. **Default** – Depends on how the driver option *Disable inline resources* is specified. If Disable inline resources is Yes, Reference mode is used, and if it is No, Embed mode is used. See *Modes for managing AFP resources* on page 22 for more information about the other modes.
.Map file (Resource)	The path to the map file for overlays and images. For example: `data/tables/mapfile.txt` This file is optional, and the settings in the file override any other settings specified for overlay and image resources. See *Mapping images and overlays* on page 33 for more information.

Settings	
.Location (Resource)	The default resource directory used by the StreamServer. This directory is used in Include and Export mode.
Font	The mode used to handle font resources. Overrides the default resource mode. **Default** – Use the default resource mode. See *Modes for managing AFP resources* on page 22 for more information about the other modes.
.Default (Font)	If there is a default font in the job, this setting can be used to specify a font name (AFP character set name, e.g. `CZ123`, or AFP coded font ID, e.g. `X0MAXI2A`) for the default font. Normally you use the file `afp2wfnt.map` to specify which names to use for the fonts. See *Mapping fonts and code pages* on page 27. You can also use this field to enter the FGID (Font Global Identifier) that applies to all generated fonts.
.Technology (Font)	Technology used for generating font resources. Applicable to modes Embed and Export, and ignored for all other modes. See *Generating font resources* on page 25.
.Content (Font)	N/A
.Location (Font)	The font resource directory used by the StreamServer. This directory is used in Include and Export mode, and overrides `.Location (Resource)`.
Code page	The mode used to handle code page resources. Overrides the default resource mode. **Default** – Use the default resource mode. See *Modes for managing AFP resources* on page 22 for more information about the other modes.

Settings	

.Default (Code page)	Overrides the auto generated code page name for the default code page. The default code page is the code page specified on the connector, or IBM CP 500 (StreamServe code page name) if no code page is specified.
	For example:
	`T1LATIN1`
	See also *Fonts and code pages* on page 25.
	You can also specify multiple code pages. If the character is not found in the first specified code page, the character is searched in the next following, and so on.
	Use the following syntax
	`<code page> -SYSCP[<system code page>]`
	and separate the code pages with a semicolon.
	For example:
	`T1000852 -SYSCP[IBM CP 852];T1000500 -SYSCP[IBM CP 500];T1000851 -SYSCP[IBM CP 851]`

.Technology (Code page)	The technology used for generating code page resources. Applicable to modes Embed and Export, and ignored for all other modes. See *Generating code page resources* on page 27.
.Content (Code page)	N/A
.Location (Code page)	The code page resource directory used by the StreamServer. This directory is used in Include and Export mode, and overrides `.Location (Resource).`
Image	The mode used to handle image resources. Overrides the default resource mode. **Default** – Use the default resource mode. See *Modes for managing AFP resources* on page 22 for more information about the other modes.
.Default (Image)	N/A
.Technology (Image)	Technology used for generating image resources. Applicable to modes Embed, Wrap, and Export. Ignored for all other modes. See *Generating image resources* on page 31.
.Content (Image)	Color mode for generating images. Applicable to modes Embed, Wrap, and Export. Ignored for all other modes. See *Generating image resources* on page 31.
.Location (Image)	The image resource directory used by the StreamServer. This directory is used in Include and Export mode, and overrides `.Location (Resource).`

Settings	
Overlay	The mode used to handle overlay resources. Overrides the default resource mode. **Default** – Use the default resource mode. See *Modes for managing AFP resources* on page 22 for more information about the other modes.
.Default (Overlay)	N/A
.Technology (Overlay)	Technology used for generating overlay resources. Applicable to modes Embed, Wrap, and Export, and ignored for all other modes. **Vector** – Generates vector based overlays. **Raster** – Generates raster based overlays.
.Location (Overlay)	The overlay resource directory used by the StreamServer. This directory is used in Include and Export mode, and overrides `.Location (Resource).`
Formdef	The mode used to handle formdef resources. Overrides the default resource mode. **Default** – Use the default resource mode. See *Modes for managing AFP resources* on page 22 for more information about the other modes.
.Default (Formdef)	The name of the generated formdef. For example: `FM2UP`
.Location (Formdef)	The formdef resource directory used by the StreamServer. This directory is used in Include or Export mode, and overrides `.Location (Resource).`
SSI headers	Enables/disables generation of SSI headers. SSI headers can be used in OS/390 environments. The SSI headers contain AFP transparent information that the Print Server can use for tailoring the printing process.
.Key (SSI *n*)	The key of SSI header number *n*. Any string can be used. This is the key that the Print Server uses to identify a specific SSI header.
.Value (SSI *n*)	The value of SSI header number *n*. Any string can be used.

Settings	
System text code page	The EBCDIC code page to use for NOP comments and TLE information. For example: `IBM CP 278` If no System text code page is specified, the code page specified for the connector is used. If the code page specified for the output is not an EBCDIC code page, you must specify an EBCDIC code page for TLE information and NOP comments.
TLE code page	The EBCDIC code page to use for TLE information. Leave this empty if you want to use the same code page as for NOP comments.
Disable inline resources	Disables the use of inline resources in the AFP data stream file. When selected, only the Reference mode can be used.
Disable mCF-2	Disables Map Coded Font function 2 (MCF2). Select this option if the output device only supports MCF1 and not MCF2.
Disable mDR	Disables Map Data Resource (MDR). Select this option if `.Technology (Font)` is Open Type, and if the printer does not support MDR. If `.Technology (Font)` is not Open Type, MDR is disabled automatically.
Disable n-up	Disables AFP n-up for side-by-side printing. Select this option if you want to use the Sheet Layout n-up functionality instead of the AFP n-up definition for side-by-side printing. For example, if the Sheet Layout has two A4s on one A3 Landscape, and you disable n-up, the A4 sheets are merged on one Landscape A3 sheet. The Print Server receives one A3 sheet, and not two A4 sheets positioned side by side.
Front pages only	Overrides duplex printing.
Disable Image Background	Some printers cannot handle image transparency correctly. Instead of printing an image with a transparent background, a black box is printed. If you disable the image background, the image is printed as a transparent image, but the opaque function is lost.
Disable Automatic Orientation	Disables the automatic orientation of logical pages on the sheet.For example, if you use partition rotation to place a logical page with landscape orientation onto a physical sheet with portrait orientation, the physical sheet will still have portrait orientation.
Disable BCOCA	Disables the Bar Code Object Content Architecture. This can be useful if your printer does not support BCOCA, or if you want to allow barcodes on a server or printer that does not support BCOCA.

Settings	
Max Record Length	The maximum record length for AFP structured fields. Which option to select depends on the capabilities of the printer. If you select None, 32 KB is used by default.
Halftone Method	The halftone method used to convert color and grayscale images to black and white. Available methods are Floyd Steinberg and Ordered Dither. If you select Default, Ordered Dither is used.
Halftone Size	The size of the halftone matrix. Default is an 8x8 matrix.
Halftone Gamma	The halftoning gamma value.
Page Group Scope	The PageGroup scope. **Default** – Use the Document definition to set the scope of the PageGroup. **Envelope** – Use the envelope definition to set the scope of the PageGroup.

Document Begin AFP driver settings

Settings	
Before document comment	A comment that will be included before the document. Any string is accepted. The comment is translated into a NOP object in the AFP data stream. You can include metadata in the string (see *Assigning metadata values to comments and TLE indexes* on page 37).
Document comment	A comment that is included in the document. Any string is accepted. The comment is translated into a NOP object in the AFP data stream. You can include metadata in the string (see *Assigning metadata values to comments and TLE indexes* on page 37).
Document name	The name of a specific BNG (Begin Named Page Group) in the AFP print file. This name corresponds to the StreamServe Document. The name can be built up using several variables, for example the Document number and user ID. If left empty, the default naming convention (`D0000001`, `D0000002`, ...) is used.
End document comment	A comment that is included at the end of the document. Any string is accepted. The comment is translated into a NOP object in the AFP data stream. You can include metadata in the string (see *Assigning metadata values to comments and TLE indexes* on page 37).

Settings	
TLE indexes	Enable/disables generation of TLE indexes. If the PageOUT output contains bookmarks and TLE indexes is enabled, page level TLE indexes are generated automatically.
.Key (TLE *n*)	The key of TLE archiving index number *n*. Any string can be used.
.Value (TLE *n*)	The value of TLE archiving index number *n*. Any string can be used (maximum 250 characters). You can include metadata in the string (see *Assigning metadata values to comments and TLE indexes* on page 37).

Page Begin AFP driver settings

Settings	
Before Page Comment	A comment that will be included before the page. Any string is accepted. The comment is translated into a NOP object in the AFP data stream. You can include metadata in the string (see *Assigning metadata values to comments and TLE indexes* on page 37).
Page comment	A comment that is included in the page. Any string is accepted. The comment is translated into a NOP object in the AFP data stream. You can include metadata in the string (see *Assigning metadata values to comments and TLE indexes* on page 37).
Page TLE indexes	Enable/disables generation of TLE indexes for StoryTeller and PageOUT output. If the PageOUT output contains bookmarks and TLE indexes is enabled, page level TLE indexes are generated automatically.
.Key (Page-TLE *n*)	The key of TLE archiving index number *n*. Any string can be used.
.Value (Page-TLE *n*)	The value of TLE archiving index number *n*. Any string can be used (maximum 250 characters). You can include metadata in the string (see *Assigning metadata values to comments and TLE indexes* on page 37).

Configuration file AFP driver settings

In the AFP driver configuration file `afp.drs` you can configure user defined halftone matrix and halftone transfer curve. This file is available from the Device Tool. See the *Device Tool* documentation for information on how to configure driver files.

Note: Configuring user defined halftone matrices requires expert knowledge in this matter.

Example 1 *User defined halftone matrix*

```
// The sample of user halftone matrix
Halftone 8 2.0
HalftoneMethod "ordered dither"
HalftoneMatrix 64
0 128 32 160 8 136 40 168
192 64 224 96 200 72 232 104
48 176 16 144 56 184 24 152
240 112 208 80 248 120 216 88
12 140 44 172 4 132 36 164
204 76 236 108 196 68 228 100
60 188 28 156 52 180 20 148
252 124 220 92 244 116 212 84
End
```

Example 2 *User defined grayscale transfer curve*

```
// The sample of user Grayscale transfer curve
HalftoneTransfer 256
  0   4   9  12  15  17  19  21  22  24  25  27  28  29  30  32
 33  34  35  37  38  39  40  42  43  45  46  48  49  50  52  53
 55  56  58  59  61  62  64  65  67  68  70  71  72  73  75  76
 77  78  79  80  81  82  83  83  84  85  86  87  88  89  90  91
 92  93  94  95  96  98  99 100 102 103 104 106 107 108 110 111
112 113 115 116 117 118 120 121 122 123 124 125 126 127 128 129
130 131 132 133 134 135 136 137 138 138 139 140 141 142 143 144
145 146 147 148 149 150 151 152 153 154 155 156 157 157 158 159
160 161 162 163 163 164 165 166 167 168 168 169 170 171 171 172
173 174 175 175 176 177 178 179 180 181 182 183 184 185 186 187
188 189 190 191 192 193 194 195 196 196 197 198 199 200 200 201
202 202 203 204 204 205 206 206 207 207 208 209 209 210 210 211
212 212 213 213 214 214 215 215 216 217 217 218 218 219 220 220
221 221 222 223 223 224 225 226 226 227 228 229 230 230 231 232
233 234 235 235 236 237 238 239 239 240 241 242 243 243 244 245
```

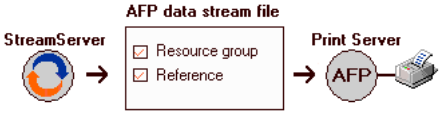
```
246 246 247 248 248 249 249 250 251 251 252 252 253 253 254 255
End
```

Modes for managing AFP resources

The AFP driver can use several modes to handle resources. You can specify a default mode used to handle the resources, and then for each resource type specify other options that override the default options.

Embed mode

The Embed mode is the default mode. It converts the original resources to AFP resources, and wraps the AFP resources into the resource group in the AFP data stream file.

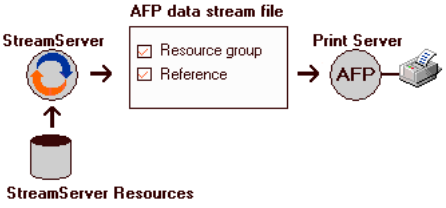


The following steps describe a simple scenario:

1. The StreamServer finds a resource on a page in the original document.
2. The StreamServer generates an AFP resource with a unique name, and wraps the AFP resource into the resource group in the AFP data stream file.
3. The StreamServer adds the unique name as a reference on the corresponding page definition in the AFP data stream file.
4. The StreamServer continues to generate and wrap AFP resources and add references for all the resources it finds.
5. The Print Server reads the AFP data stream file.
6. When the Print Server finds a reference in the AFP data stream file, it retrieves the corresponding resource from the resource group in the AFP data stream file.

Include mode

The Include mode enables the use of external resources (3rd party or modified resources). This mode requires that the external resources are included in the resources storage used by the StreamServer.



The following steps describe a simple scenario:

1. The StreamServer finds a resource on a page in the original document.
2. The StreamServer generates a unique name for the resource, retrieves the corresponding AFP resource from the resources storage, and wraps it into the resource group in the AFP data stream file.
3. The StreamServer adds the unique name as a reference on the corresponding page definition in the AFP data stream file.

4 The StreamServer continues to retrieve and wrap AFP resources, and add references for all resources it finds.

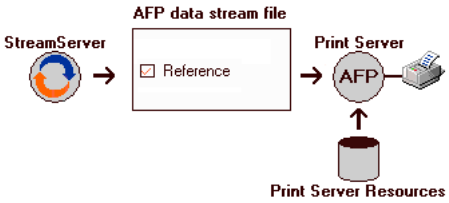
5 The Print Server reads the AFP data stream file.

6 When the Print Server finds a reference in the AFP data stream file, it retrieves the corresponding resource from the resource group in the AFP data stream file.

Note: The names of the resources in the resources storage must correspond to the resource names generated by the StreamServer.

Reference mode

The Reference mode can be used to optimize conversion speed, the size of the AFP data stream file, and resource loading on the printer. This mode requires that the resources are included in the resources storage used by the Print Server..



The following steps describe a simple scenario:

1 The StreamServer finds a resource on a page in the original document.

2 The StreamServer generates a unique name for the resource, and adds this name as a reference on the corresponding page definition in the AFP data stream file.

3 The StreamServer continues to add references for all resources it finds.

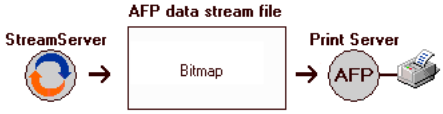
4 The Print Server reads the AFP data stream file.

5 When the Print Server finds a reference in the AFP data stream file, it retrieves the corresponding resource from the resources storage.

Note: The names of the resources in the resources storage must correspond to the resource names generated by the StreamServer.

Merge mode

The Merge mode merges images and overlays as bitmaps directly on a page definition. This mode can be used for unique images/overlays that are not shared by several page definitions.



The following steps describe a simple scenario:

1 The StreamServer finds a resource on a page in the original document.

2 The StreamServer generates a bitmap, and merges the bitmap on the corresponding page definition in the AFP data stream file.

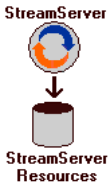
3 The StreamServer continues to generate and merge bitmaps for all resources it finds.

4 The Print Server reads the AFP data stream file.

5 When the Print Server finds a bitmap, it uses this bitmap in the output.

Export mode

The Export mode can be used to generate AFP resource files. The generated files can be uploaded to the resources storage used by the Print Server, and later be used with the Reference mode. The resource files can also be modified, and later be used with the Include mode.



The following steps describe a simple scenario:

1 The StreamServer finds a resource on a page in the original document.

2 The StreamServer generates an AFP resource with a unique name, and exports the resource to a file in the resources storage used by the StreamServer.

3 The StreamServer continues to generate and export AFP resources for all resources it finds.

Ignore mode

The Ignore mode is only applicable to image, overlay, and formdef resources. It can be used during development to ignore temporary problems with resources. It can also be used during production where the corresponding resources are pre-printed on paper.

The following steps describe a simple scenario:

1 The StreamServer finds a resource on a page in the original document.

2 The StreamServer ignores the resource, and adds no resource information to the AFP data stream file.

3 The StreamServer continues to ignore all resources it finds.

4 The Print Server reads the AFP data stream file.

5 The Print Server finds no resources, and no resources are included in the output.

Fonts and code pages

Font output from PageOUT can be handled by the AFP driver using the following modes:

- Embed
- Include
- Reference
- Export

See *Modes for managing AFP resources* on page 22 for information about the above driver options.

Auto generated font names

By default, the AFP driver creates sequential names for fonts in the same job. The first font found in a job is named *prefix*000000 (where *prefix* is CZ for outline fonts and C0 for raster fonts), the next is named *prefix*000001, and so on. This method can only be used if the AFP driver runs in Embed mode for font resources.

Specified font names

If the AFP driver runs in Include, Reference, or Export mode for font resources, the font names must not change between jobs. This means auto generated font names cannot be used. You must therefore manually map the fonts to unique names. You can use any name for a font as long as it has the right prefix (CZ for outline fonts and C for raster fonts) and is less than 9 characters long (including the prefix). See *Mapping fonts and code pages* on page 27 for more information on how to name fonts.

Generating font resources

To generate font resources Embed and Export mode) you must use the driver setting **.Technology (Font)** to specify which technology to use to generate font resources.

Font technologies	
Outline	Embeds Adobe Type 1 fonts in AFP outline font resources.
CID Outline	Embeds Adobe CID fonts in AFP outline font resources.
Raster	Generates AFP raster font resources. The font resolution is determined by the **Resolution** driver setting. If you specify 240 dpi, fixed raster metrics is used. If you specify a higher resolution, relative raster metrics is used.
Relative Raster	Generates AFP raster font resources using relative raster metrics.

Font technologies	
Fixed Raster	Generates AFP raster font resources using fixed raster metrics. The font resolution is determined by the **Resolution** driver setting.
OpenType	Embeds OpenType and TrueType fonts as data resources rather than traditional AFP font resources. This new technology is not supported by most printers.
TT Outline	Embeds TrueType fonts in AFP outline font resources. Used only for OCE printers that do not support Type 1 fonts.

Fixed raster metrics compensation

If a fixed raster metrics technology is used to generate font resources, the width of the printed characters can be affected when, for example, printing long right-aligned texts, or when the character format changes (e.g. from normal to bold). The driver compensates for this effect if the fonts are mapped to ReadFonts in the file `afp2wfnt.map`. If the fonts are mapped to Width Table fonts, no compensation is made. See *Mapping fonts and code pages* on page 27 for more information about `afp2wfnt.map`.

Characters included in a font resource

When running the AFP driver in Embed mode for font resources, only the characters that are used are included in the font resource wrapped into the resource group.

When running the AFP driver in Export mode for font resources, all characters from the code page specified for the font are included in the exported font resource. This means if you use exported font resources when running the AFP driver in Include mode, all characters from the code page specified for the font are included in the font resource.

Code pages

By default, the code page specified on the output connector is used for all fonts. If no code page is specified, `IBM CP 500` is used.

Auto generated code page names

A code page is normally uniquely named by resolving the StreamServe code page name. For example, `IBM CP 500` is named `T1000500`. In this case, the auto generated code page name can be used in all modes for code page resources (Embed, Include, etc.)

A code page that cannot be named uniquely by resolving the StreamServe code page name is given a sequential number, for example `T1000000`. In this case, the auto generated code page name can only be used if the AFP driver runs in Embed mode for code page resources.

Specified code page names

You can use the AFP driver setting **.Default (Code page)** to explicitly name the default code page. The name you enter in this field overrides the auto generated code page name. The code page name must begin with `T1`, and be less than 9 characters long.

You can also use the file `afp2wfnt.map` to name the default code page, as well as any other code pages used in the job. See *Mapping fonts and code pages* on page 27 for more information.

Unicode

Unicode is applicable only if the AFP driver runs in Embed mode for code page resources. If you specify unicode (StreamServe name `Unicode (UCS-2)`) as the default code page on the output connector, the auto generated AFP code page name is `T11200`.

Code page for TLE information and NOP comments

External applications that read TLE information and NOP comments expect EBCDIC code pages (StreamServe code page name `IBM CP nnn`). This means if the code page specified for the output is not an EBCDIC code page, you must specify an EBCDIC code page for the TLE information and NOP comments. The AFP driver setting **System text code page** applies to both TLE information and NOP comments. If you need a separate code page for TLE information, you can use the AFP driver setting **TLE code page**.

Generating code page resources

Normally you will use the default technology to generate code page resources (Embed and Export mode).

If you need to produce double byte output not only for printing, but also for archiving, searching, etc., you must set the AFP driver option **.Technology (Code page)** to **Coded Font for Double Byte**. This means the AFP driver converts double byte code pages to a set of single byte AFP code page resources and AFP font resources, together with AFP coded fonts.

Mapping fonts and code pages

When using external font and code page resources, i.e. when running the AFP driver in Include or Reference mode, you must make sure the resource names in the AFP data stream file are the same as the names of the external resources. You must therefore map the fonts and code pages used in PageOUT to the corresponding external resources. To do this, you edit the AFP driver file `afp2wfnt.map` using the Device Tool. You can also use the `afp2wfnt.map` file to specify options for a specific font or code page. See the *Device driver tools* documentation for information on how to edit driver files.

The settings in the `afp2wfnt.map` file override the corresponding AFP driver GUI settings for fonts and code pages. A standard entry for a font has the following format in the `afp2wfnt.map` file:

```
Font "TTF_name"
ReadFont "TTF_file" SelectPrefix "TTF_name" Codepage "Source_CP"
Select "AFP_font" Codepage "AFP_CP"
```

Example 3 *Font entry in afp2wfnt.map*

```
Font "Arial"
ReadFont "ARIAL.TTF" SelectPrefix "Arial" Codepage "Ansi"
Select "CZH200" Codepage "T1000870"
```

afp2wfnt.map parameters	
TTF_name	The original TrueType font used in PageOUT. The name must contain all used flags, e.g. bold or italic. You must use underscores as separators. For example `Arial_bold_italic`
TTF_file	The TrueType font file read by the StreamServer.
AFP_font	The name to use for the AFP font resource.
Source_CP	The source code page.
AFP_CP	The name to use for the AFP code page resource.
-Technology[*Technology_font*]	The font technology to use to generate font resources. You can specify the following *Technology_font* options: • CID • OUTLINE • TTOUTLINE • RASTER • RELATIVERASTER • FIXEDRASTER • OPENTYPE. See *Generating font resources* on page 25 for more information about these options.

afp2wfnt.map parameters	
-Technology[*Technology_CP*]	The code page technology to use to generate code page resources. You can specify the following *Technology_CP* options: • DEFAULT • CODEDFONT See *Generating code page resources* on page 27 for more information about these options.
-mode	The mode to use for the font/code page resource. You can specify the following modes: • INCLUDE • REFERENCE • EXPORT • EMBED See *Modes for managing AFP resources* on page 22 for more information about these options.
Location_font	The location of the font resource. Used only if *mode* is INCLUDE or EXPORT. If *mode* is INCLUDE, the resource will be retrieved from this location. If *mode* is EXPORT, the resource will be exported to this location.
Location_CP	The location of the code page resource. Used only if *mode* is INCLUDE or EXPORT. If *mode* is INCLUDE, the resource is retrieved from this location. If *mode* is EXPORT, the resource is exported to this location.
-SYSCP[*System_CP*]	System code page to use for a font, where System_CP is the StreamServe name of the code page to use. For example: -SYSCP[ISO 8859-1]

Mode

To specify whether to export, include, embed, or reference a specific font or code page resource, you must add the *mode* parameter to the Select row.

Example 4 *Embedding a font and code page resource*

```
Select "CZH200 -EMBED" Codepage "T1000870 -EMBED"
```

Export/Include path

To be able to export or include a font or code page resource, you must add the *Location_font* and *Location_CP* parameters to the Select row.

Example 5 *Including a font and code page resource*

```
Select "CZH200 -INCLUDE Data\Fonts\*.OLN" Codepage "T1000870
-INCLUDE Data\Fonts\*."
```

Font sizes for raster fonts

You can specify which raster font to use for a specific font size.

Example 6 *An entry specifying which font to use according to font size*

```
Font "Arial"
ReadFont "ARIAL.TTF" SelectPrefix "Arial" Codepage "Ansi"
Size 7 Select "COH20080"
Size 8 Select "COH20080 -INCLUDE data\fonts\*.300
Size 9 Select "COH20090.300"
Size 10 Select "COH20000.300"
```

Font technology

You can use the parameter -Technology [*Technology_font*] to specify which technology to use when generating a specific font.

Example 7 *Entries specifying font technology*

```
Font "Arial"
ReadFont "ARIAL.TTF" SelectPrefix "Arial" Codepage "Ansi"
Select "COH20000 -EXPORT -Technology [RASTER] data\fonts\*.300"
Font "Times_New_Roman"
ReadFont "TIMES.TTF" SelectPrefix "Times New Roman" Codepage "Ansi"
Select "CZN200 -EXPORT -Technology [CID] data\fonts\*.OLN"
```

System code page

You can use the parameter -SYSCP[*System_CP*] to specify a specific system code page for a font. This will override the default system code page specified for the output connector.

Example 8 *An entry specifying a system code page for a font*

```
Font "MS_Mincho"
ReadFont "MSMINCHO.TTF" SelectPrefix "MS_Mincho" Codepage "Ansi"
Select "CZM001" Codepage "T1000950 -SYSCP[BIG5]"
```

Images and overlays

Image and overlay output from PageOUT can be handled by the AFP driver using the following modes:

- Embed
- Include
- Reference
- Export
- Merge
- Ignore

See *Modes for managing AFP resources* on page 22 for more information about the above driver options.

Auto generated image and overlay names

The AFP driver auto generates resource names for images and overlays as *PrefixName*, where *Prefix* is S1 for images and O1 for overlays, and *Name* is the name of the image/overlay used in PageOUT. For example:

```
S1LOGO
O1SLIP
```

The resource name, including prefix, can be up to eight characters long. This means that if the original image/overlay contains more than six characters, the name will be truncated. For example, an image called logoSWE in PageOUT will generate the image resource name S1LOGOSW.

Auto generated resource names are normally used when running the AFP driver in Embed and Merge mode for image and overlay resources.

Specified image and overlay names

When using external image and overlay resources (Include and Reference mode), you must make sure the resource names in the AFP data stream file are the same as the names of the corresponding external resources. This means you can only use auto generated resource names if you first export the resources, and then use the exported resources as external resources.

If you use other resources (e.g. 3rd party generated image and overlay resources) as external resources, you must map the original images and overlays used in PageOUT to the corresponding external resources. See *Mapping images and overlays* on page 33 for more information on how to name images and overlays.

Generating image resources

When generating image resources (Embed, Export and Merge mode), you must specify the resolution and color range for the image resources. You can also specify different modes for generating color images.

Resolution

The AFP driver setting **Resolution** specifies the resolution of the generated images. This setting also sets the resolution of all generated overlays and fixed raster fonts.

Color settings

The AFP driver settings **Color** and **.Technology (Image)** specifies whether to generate black and white, or color images. To generate color images, you must select **Color** > **Extended** and **.Technology (Image)** > **IO Image** or **IO Image Compressed**.

IOCA settings

The IOCA settings (AFP driver setting **.Content (Image)**) apply to color images. The following modes are available:

- IOCA FS11 (**.Content (Image)** > **RGB**)
- IOCA FS45 (**.Content (Image)** > **CMYK**)
- IOCA FS10 (**.Content (Image)** > **B/W**)

The mode to select depends on the capabilities of the printer. The default mode is IOCA FS45, and is normally used for printing. IOCA FS11 is normally used for AFP viewers.

Generating overlay resources

When generating overlay resources (Embed, Export, and Merge mode), you must specify the resolution and color range for the overlay resources. You can also specify different GOCA modes for generating vector graphics.

Resolution

The AFP driver setting **Resolution** specifies the resolution of the generated overlays. This setting also sets the resolution of all generated images and fixed raster fonts.

Color settings

The AFP driver setting **Color** specifies whether to generate black and white, gray scale, or color overlays. The option to select depends on the capabilities of the printer. The following options are available:

- **None** – All colors are mapped to black and white.
- **Yes** – All colors are mapped to a limited set of colors.
- **Extended** – All colors are mapped to the RGB 0-255 color model.
- **Grayscale** – All colors are mapped to 256 shades of gray.

GOCA settings

The AFP driver setting **GOCA** applies to vector graphics in the overlay, and also to vector graphics drawn directly in the PageOUT tool. Which option to select depends on the capabilities of the printer. The following options apply:

- **None** – GOCA is not used to generate vector graphics. Only vertical and horizontal lines are presented in the output.
- **Yes** – Algorithms with simple GOCA objects are used to generate vector graphics. Enables printing of free lines, polygons, ovals, and round corners.
- **Extended** – The extended GOCA set is used to generate vector graphics without GOCA fillets. Enables printing of free lines, polygons, ovals, and round corners.
- **Full** – The full GOCA set is used to generate vector graphics with GOCA fillets. Enables printing of free lines, polygons, ovals, and round corners.
- **Raster All** – All vector graphic objects on pages or overlays are rasterized. Enables printing of free lines, polygons, ovals and round corners as raster images.
- **Raster Patterns** – All vector graphic objects with patterns are rasterized. GOCA supports only a limited set of patterns. This option allows rasterizing of objects filled by unsupported patterns.

Mapping images and overlays

When using external image and overlay resources (Include and Reference mode), you must make sure the resource names in the AFP data stream file are the same as the names of the external resources. This means you must map the original images and overlays used in PageOUT to the corresponding external resources.

To do this, you create a map file that describes how to map original images and overlays to external resources. Then you add the map file to the AFP driver configuration. The map file settings override the corresponding AFP driver GUI settings for images and overlays.

To create a map file

1. Create a new table resource in a resource set
2. Rename the table resource. For example to `resourcemap`.
3. Open the table resource.
4. For each image/overlay you want to map, add a new line using the following syntax:

   Item TAB *Type* TAB *Name* TAB *Mode* TAB *Location*

   The parameters are described in the table below.
5. Save and close the table resource.

Parameters	
Item	The name defined for the overlay/image in PageOUT.
Type	PSEG for images and OVERLAY for overlays.
Name	The AFP resource name to use for the image/overlay, for example 01SLIPEN. Image resources have the prefix S1, and overlay resources have the prefix O1. The resource name, including the prefix, can be up to eight characters long.
Mode	The mode to use for the image/overlay resource. The following options apply: • INCLUDE • REFERENCE • EXPORT • EMBED • IGNORE • MERGE See *Modes for managing AFP resources* on page 22 for more information about these options.
Location	The location of the image/overlay resource. Used only if *Mode* is INCLUDE or EXPORT. The location is the path to the file that contains the resource, for example AFPRESOURCES/01SLIPEN.OVL. If *Mode* is INCLUDE, the resource is retrieved from this location. If *Mode* is EXPORT, the resource is exported to this location.

To add the map file to the AFP driver configuration

The map file you created is exported from the resource set to <*export*>\data\tables. You must specify this path when you configure the device driver settings.

1 Open the Runtime Connector settings dialog for the output connector that delivers the AFP output.

2 Select **Job Begin** and **Device Driver Settings**.

3 In **.Map file (Resource)**, enter the path to the map file, for example:

```
data\tables\resourcemap
```

Examples

Example 9 *Mapping images and overlays*

In this example, the image `Logo.gif` and the overlay `Slip.lxf` are used in PageOUT. The image and overlay are added to the AFP data stream file using Include mode. The image resource is retrieved from the file `S1LOENG.PSG`, and is named `S1LOENG`. The overlay resource is retrieved from the file `O1SLIPEN.OVL`, and is named `O1SLIPEN`. Both resource files are stored in the resource directory `AFPRESOURCES` in the Export directory of the Project.

The map file looks like this:

```
//!CodePage UTF8!
Slip    OVERLAY    01SLIPEN    INCLUDE    AFPRESOURCES/01SLIPEN.OVL
Logo     PSG        S1LOEN      INCLUDE    AFPRESOURCES/S1LOEN.PSG
```

Form definitions

A form definition (formdef) is a resource object that defines the layout and characteristics of logical pages on a sheet. A form definition is normally created using the Sheet layout editor. If you do use the Sheet layout editor, a default form definition is generated. This default form definition is based on the layout defined in the PageOUT tool.

Formdef output can be handled by the AFP driver using the following modes:

- Embed
- Include
- Reference
- Export
- Ignore

See *Modes for managing AFP resources* on page 22 for more information about the above driver options.

Variables and metadata

You can assign variables to the device driver settings at both Job Begin and Document Begin.

To assign a variable to a device driver setting

You cannot enter a variable into the property field, you must use the Alias method to enter the variable.

1. On the Device Driver Settings tab, at Job Begin or Document Begin, select the device driver property. The selected property is displayed as an Alias option at the bottom of the Runtime Connector Settings dialog box.
2. Select **Variable** and enter the variable in the **Variable** field.

Example 10 Assigning a variable to .Value (TLE 1)

In this example, the variable `$zip` is assigned to the property `.Value (TLE 1)`.

1. Select **.Value (TLE 1)** on the Device Driver Settings tab at Document Begin. `Alias for Option: .Value (TLE 1)` is displayed at the bottom of the Runtime Connector Settings dialog box.
2. Select **Variable** and enter `$zip` in the **Variable** field.

Assigning metadata values to comments and TLE indexes

The text strings that specify TLE values and NOP comments can include metadata generated for post-processing purposes (bundling, sorting, enveloping, etc.). You include the metadata as variables using the following format:

`%{MetadataName}`

For example, to add the number of sheets in a document to a comment, you enter `%{NumSheets}` in the comment field.

Page identification parameters included in the generated AFP data stream file

The generated AFP data stream files contains a number of page identification parameters that can be used by the Print Server. The following parameters are added as a NOP comment below each Begin Page node in the AFP data stream file:

- SIDE – Front (F) or back (B).
- DOCNR – The document number.
- DOCPG – The page number in a document.
- DOCSH – The sheet number in a document. This is only specified if a sheet layout resource is used.
- JOBPG – The page number in a job.
- JOBSH – The sheet number in a job. This is only specified if a sheet layout resource is used.

Optimizing performance

Page size

The number of bytes on each logical page affect performance. The page size is mainly determined by page density and complexity, for example image scaling and resolution.

Page density

The page density affects performance more than the number of bytes on the page. Reduce the page density as much as possible.

Resolution

If possible, make sure the resolution of the input data is the same as the printer resolution.

Scaling

If possible, make sure that images do not need to be scaled by the printer control.

Fonts

Reduce the number of fonts used in a job. Switching between different fonts can affect printing speed. Use fonts that are stored in the printer if possible.

Redundancy

Avoid defining fonts, text orientation, and positioning unless it is required. Avoid unnecessary blank characters and lines that overlap. For uncompressed images, reduce the number of areas of white space if possible.

AFP objects

Reduce the number of objects, structured fields, and controls. Try to avoid switching between different AFP object types.

Representation

Use efficient representation, for example solid lines instead of dashed, and group data that uses the same font. Use compressed images instead of uncompressed, and where possible use IOCA images instead of GOCA images.

Optimizing printer efficiency for AFP throughput

There are several things to consider when optimizing printer efficiency for AFP output:

- Retain resources in the printer storage.
- Use overlays for sections that are common to several pages.
- Avoid creating many short jobs. Job initiation and termination can affect printer performance.
- Minimize the need to switch between simplex and duplex.
- Minimize the need to switch between different input bins.

Upgrading an AFP (StreamServe 3.x) driver

Normally you only have to select the AFP driver on the connector to upgrade an AFP (StreamServe 3.x) driver to an AFP driver (the new AFP driver described in this document). However, if the 4.x Project Tool Project is using local INI files, you must upgrade the driver manually.

Note: You do not have to update the driver manually in Design Center Projects.

To upgrade an AFP (StreamServe 3.x) driver

1. Open <*strs_project_path*>/Init/Devices1.txt and add the following line:

   ```
   AFP2   AFP2   AFP2_opt.txt   afp2.drs    FILE:AFP   2    1
   ```

2. Copy <*strs_path*>/Init/AFP2_opt.txt to <*strs_project_path*>/Init
3. Copy <*strs_path*>/Drivers/afp2.drs to <*strs_project_path*>/Drivers
4. Copy <*strs_path*>/Drivers/afp2wfnt.map to <*strs_project_path*>/Drivers
5. Configure the AFP driver to match the old options.

HTML

The HTML driver is used when generating HTML output.

Runtime Job begin settings	
Home URL	Specify a home URL, for example: `http://www.streamserve.com` This link will be available in the banner section.
Search URL	Specify a search URL, for example: `http://www.google.com` This link will be available in the banner section.
Create banner	Select to create a page banner section. In this section it will be possible to navigate through the generated HTML files by selecting `Next` and `Previous`.
Create index	Select to create an index. This index will be available in the banner section.
Create continuos	**Yes:** Create one html file for all pages. **No:** Create one html file per page.
Page title	Page title.
Page description	Sets the abstract that a client sees in the query results. For example, if you enter `Available colors` in this field, the corresponding meta tag in the HTML output will look like this: `<meta name="description" content="Available colors">`
Page keywords	Keywords a client can search for in a query. You can separate keywords with commas. For example, if you enter `Red, Blue, Green` in this field, the corresponding meta tag in the HTML output will look like this: `<meta name="keywords" content="Red, Blue, Green">`

Runtime Job begin settings	
Banner height	Applicable only if CreateBanner is set to Yes. The default BannerHeight is 30 (px). With this setting the output page will start with the banner at the top, and a line 30px below the top. The rest of the output will be positioned below the line. If you increase/decrease BannerHeigh by for example 5, the line and the rest of the document will be moved downwards/upwards 5px – the banner section will not be moved.
Scaling factor	Scale the output. A scale factor of 1 represents 100% of the original size of the included objects (images, text, graphics, etc.).

Image output folder	By default, all images are exported from the resource set to `data\images`. In the HTML output, the source reference will point to this directory, for example:
	`<img ... src="data/images/star.jpg">`
	You can use `ImageOutputFolder` to specify an alternate image directory. You can specify an absolute path, or a path relative to the export directory. In the HTML output, the source reference will point to this directory, for example:
	`<img ... src="C:/images/star.jpg">`

Enable PNG	The HTML output may contain `*.JPG`, `*GIF`, and `*.PNG` images. Any other format will be converted to `*.JPG`. With EnablePng you specify how to handle `*.PNG` images. **YES:** The `*.PNG` image will be handled as it is. **NO:** The `*.PNG` image will be converted to `*.JPG`.

Runtime Job begin settings	

Copy linked images	The PageOUT configuration may include linked images that are not included in any resource set. For example, if you create an lxf overlay using a standalone Overlay Editor, and link images to the overlay, these linked images will not be added to any resource set. With `CopyLinkedImages` you specify how to handle the linked images.
	YES:
	The image will be copied from the source directory to either `data\images` or the target directory specified by `ImageOutputFolder`. In the HTML output, the source reference will point to this directory, for example:
	`<img ... src="data/images/star.jpg">`
	NO:
	In the HTML output, the source reference will point to the source directory, for example:
	`<img ... src="C:/images/star.jpg">`

44 HTML

IJPDS

The StreamServe IJPDS driver generates a data stream in IJPDS format (Ink-Jet Printer Data Stream). The data stream is used in a Kodak Versamark IJPDS printing environment.

Prerequisites

To configure the IJPDS driver you must have detailed knowledge in:

- The IJPDS printing environment and the hardware configuration.
- Color management.

References

For information about the Kodak Versamark IJPDS printings system, see *Kodak Versamark IJPDS* documentation.

Known limitations

The following known limitations apply to the IJPDS driver:

- Only LXF and TIFF overlays are supported.
- Trapping is not supported.
- Knockout is not supported.
- Converting images to fonts is not supported.

Supported data stations

Kodak Versamark data stations CD100, CD120 and CD130 are supported. The supported IJPDS job file records are listed in the table below.

Record		Data station version
BOX	Fill a Rectangular Area	All versions
CBM	Compressed Bitmap	All versions
CCD	Compressed Character Definition	Version 1.16 and later
CDM	Character Definition with Metrics	Version 1.13 and later
EFF	End Fixed Files	All versions
EOJ	End of Job	All versions
FAR	Font Assignment Record	All versions
GFF	Go to Fixed File	All versions
IBM	Image Bitmap	All versions

Record		Data station version
JC2	Job Control Record 2	All versions
LFF	Load Fixed Files	All versions
NOP	No Operation	All versions
PHR	Printhead Requirements Record	All versions
RFF	Reset Fixed File	All versions
RIP	Select RIP	All versions
SDC	Start of Document and Cue	All versions
SFM	Super Font with Metrics	Version 1.13 and later
SFT	Set Font	All versions
SIL	Two-Byte Image Line	All versions
SLF	Set Logical Function	Version 1.14 and later
SOR	Set Origin	Version 1.13 and later
SPO	Set Position	All versions
SPX	Set Position Extended	Version 1.14 and later

Configuring the IJPDS driver

To specify the IJPDS driver settings you must use a configuration file in XML format. You can copy and edit one of the template files `ijpds.mono.cfg` or `ijpds.cmyk.cfg`. These files are available from the Device Tool. See the *Device Tool* documentation for information on how to use the Device Tool.

To include the files in the Project export, you should import the configuration file and the color lookup tables to the Project resource set.

To create a configuration file

1. In the Device Tool, select the **IJPDS** device.
2. Click **Duplicate**. A duplicate of the IJPDS device is added.
3. In the **Device driver files list**, double-click `ijpds.mono.cfg` or `ijpds.cmyk.cfg`. A "Yes/No" dialog box opens
4. Click **Yes**. The template opens.
5. Copy the template XML and paste it into a text editor.
6. Exit the Device Tool.
7. Edit and save the configuration file.

To configure the IJPDS driver

1. In the Platform connector configuration, select the IJPDS driver.
2. In the Runtime connector configuration, click **Job Begin** and select the **Device Driver Settings** tab.
3. Enter the name of the configuration file.

Preparing the configuration file

You specify the driver settings in the IJPDS configuration file, for example:

- Resolution
- Color lookup tables
- Color separation
- Printhead and RIP settings

For a quick reference of the properties in the configuration file, see *IJPDS driver quick reference* on page 72.

IJPDS configuration file syntax

The value of the `<property>` element can be specified in three different ways in the configuration file:

- As an element content.
- Using the `value` attribute.

- Using the `choice` attribute and a `<choices>` list.

The examples below show the different ways to configure the `x-resolution` to `300`.

Example 11 Horizontal resolution specified as an element content

```
<property name="x-resolution">300</property>
```

Example 12 Horizontal resolution specified using the value attribute

```
<property name="x-resolution" value="300"></property>
```

Example 13 Horizontal resolution specified using the choice attribute and a choices list

```
<property name="x-resolution" choice="3">
    <choices>
        <property>120</property>
        <property>240</property>
        <property>300</property>
    </choices>
</property>
```

Resolution

Resolution can differ in horizontal and vertical directions. The paper movement direction should normally have a higher value.

To configure the resolution

You configure the horizontal resolution using the `x-resolution` property and the vertical resolution using the `y-resolution` property.

Example 14 Resolution settings

In this example, the horizontal resolution is `240` DPI and the vertical resolution is `300` DPI.

```
<property name="x-resolution">240</property>
<property name="y-resolution">300</property>
```

Color management

The output color is dependent on a number of factors, such as paper quality, printing speed and the amount of ink. To achieve exact color representation you can use color lookup tables to transform the input color to output color. You can use different sets of color lookup tables to print images, text and vector graphics. Each set contains one color lookup table for each color (cyan, magenta, yellow and black).

If an individual image, for example a logotype, should be printed in a different color tone than the other images, you can use another set of color lookup tables to transform this image. To convert RGB images to CMYK you can use ICC color profiles.

You should import all color lookup tables to the resource set to include them in the export.

Using color lookup tables

In the configuration file, you configure the color lookup table sets and which set to use when printing images, vector graphics and text.

To configure a color lookup table set

1. Configure a unique ID for the color lookup table set with the `lut` property.
2. Configure the name of the:
   - Cyan color lookup table with the `lutc` property.
   - Magenta color lookup table with the `lutm` property.
   - Yellow color lookup table with the `luty` property.
   - Black color lookup table with the `lutk` property.

Example 15 Settings for color lookup table sets

In this example, two color lookup table sets with ID `6` and `7` are configured. Each set contains four color lookup tables, one for each color.

```
<property name="luts">
  <property name="lut" id="6">
    <property name="lutc">data/tables/Curve1,Lev6_c.lut</property>
    <property name="lutm">data/tables/Curve1,Lev6_m.lut</property>
    <property name="luty">data/tables/Curve1,Lev6_y.lut</property>
    <property name="lutk">data/tables/Curve1,Lev6_k.lut</property>
  </property>
  <property name="lut" id="7">
    <property name="lutc">data/tables/Curve1,Lev7_c.lut</property>
    <property name="lutm">data/tables/Curve1,Lev7_m.lut</property>
    <property name="luty">data/tables/Curve1,Lev7_y.lut</property>
```

```
    <property name="lutk">data/tables/Curve1,Lev7_k.lut</property>
  </property>
</property>
```

To configure the color lookup table sets for images, vector graphics and text

You configure the color lookup table set to use for:

- Images with the `imagelut` property.
- Vector graphics with the `vectorlut` property.
- Text with the `textlut` property.

Example 16 Settings for color lookup table sets for images, vector graphics and text

In this example, the tables in the table set with ID `6` are used for printing images. The tables in the table set with ID `7` and `8` are used for printing vector graphics and text.

```
<property name="imagelut">6</property>
<property name="vectorlut">7</property>
<property name="textlut">8</property>
```

Using specific color lookup tables for individual images

To print individual images in a different color tone, you must use an image lookup table. In this table, you specify the names of the image files and the IDs of the color lookup table sets to use. All other images are printed with the color lookup table set specified with the `imagelut` property, see *To configure the color lookup table sets for images, vector graphics and text* on page 51.

The image lookup table must have two tab-separated columns and the following syntax:

```
<Image_Filename>      <Color_Lookup_Table_Set_ID>
```

Example 17 Image lookup table

In this example, the color lookup table set with ID `10` is used to transform the `logo.jpg` file and `12` is used to transform the `photo.jpg` file.

```
logo.jpg          10
photo.jpg         12
```

To use color lookup tables for individual images

1 Create the image lookup table and import it to the resource set.

2 In the configuration file, specify the name of the image lookup table with the `imagetable` property.

Example 18 Using an image lookup table

In this example, an image lookup table named `Images.tbl` is used to transform images.

```
<property name="imagetable">data/tables/Images.tbl</property>
```

Using ICC profiles for RGB to CMYK conversion

You can use ICC color profiles for conversion of raster images to CMYK color space. You can apply both color lookup tables and ICC color profiles on raster images.

In the configuration file, you can use the rendering intent methods listed below.

Method	Description	ICC name
`Picture`	The full gamut in the image is compressed or expanded to fill the printer gamut. The gray balance is preserved, but colorimetric accuracy may not be preserved. Most suitable for photographs and images.	Perceptual
`Graphic`	Preserves the saturation of colors in the image at the possible expense of hue and lightness. Most suitable for business graphics, such as charts, where it is important that the colors contrast well with each other.	Saturation
`Proof`	The colors are unchanged, except colors that fall outside the range that the output device can handle. These colors are adjusted to the closest color that can be rendered. The white point of the image is mapped to the white point of the printer.	Relative Colorimetric
`Match`	The same applies as for proof, except that the white point is preserved.	Absolute Colorimetric

To use an ICC profile

1 Specify the name of the ICC profile with the `rgb2cmykprofile` property.

2 Specify the rendering intent method with the `renderingintent` property.

Example 19 ICC color profile settings

In this example, raster images are converted to CMYK with the ICC color profile `ijpds.icm` and the rendering intent method `picture`.

```
<property name="rgb2cmykprofile">data/tables/ijpds.icm
  <property name="renderingintent">picture</property>
</property>
```

Color separation

Each printhead prints only one color and a color separation process is used to decompose the color image into single-color layers. The process separates the image using plates which produce four halftone patterns in cyan, magenta, yellow and black. To fine-tune the color separation you can print color registration marks.

To reduce the amount of ink you can replace rich black and CMY black with black. You can also configure the total amount of ink to use.

Configuring the color a RIP should print

For monochrome printing you should configure one plate for the black color. For color printing you should configure four plates, one for each color. To specify the color a RIP should print, you assign a plate to the RIP.

For general information on how to configure RIPs, see *Using a RIP with single printhead* on page 61.

To print monochrome

1. Configure the black color and give the plate a unique ID with the `plate` property. For example:

   ```
   <property name="plate" id="0">black</property>
   ```

2. Configure a RIP group and the RIP and its properties with the `ripgroup` property.

3. Assign the plate to the RIP with the `plate` property. For example:

   ```
   <property name="plate" type="int">0</property>
   ```

Example 20 Color separation for monochrome printing

In this example, the RIP prints black on the front side of the sheet.

```
<!--Color separation plate description-->
<property name="plate" id="0">black</property>


<!--RIP group definition-->
<property name="ripgroup" id="0">
   <property name="type">frontpage</property>
   <property name="rip" id="2">
       <property name="plate" type="int">0</property>
       <property name="printheads">0</property>
   </property>
</property>
```

To print in color

1. Configure four plates, one for each color, and give each plate a unique ID with the `plate` property.
2. Configure a RIP group and the RIPs and their properties with the `ripgroup` property.
3. Assign plates using the `plate` property for each RIP in the RIP group.

Example 21 Color separation for color printing

In this example, four RIPs are used. The RIP with ID `0` prints cyan, `1` prints magenta, `2` prints black and `3` prints yellow.

```
<!--Color separation plate description-->
<property name="plate" id="0">cyan</property>
<property name="plate" id="1">magenta</property>
<property name="plate" id="2">black</property>
<property name="plate" id="3">yellow</property>

<!--RIP group definition-->
<property name="ripgroup" id="0">
   <property name="type">frontpage</property>
   <property name="rip" id="0">
     <property name="plate" type="int">0</property>
     <property name="printheads">0,0</property>
   </property>
   <property name="rip" id="1">
     <property name="plate" type="int">1</property>
     <property name="printheads">0,0</property>
   </property>
   <property name="rip" id="2">
     <property name="plate" type="int">2</property>
     <property name="printheads">0,0</property>
   </property>
   <property name="rip" id="3">
     <property name="plate" type="int">3</property>
     <property name="printheads">0,0</property>
   </property>
</property>
```

Printing color registration mark

You can print color registration marks for fine-tuning the color separation. The registration mark is 8.0 by 8.0 millimeters and id printed in a position relative the cursor starting position (the upper left corner of the IJPDS document).

To print color registration marks

You configure where to print a registration mark using the `registrationmark` property. Specify the horizontal position of the mark center with the `x` property, and the vertical position with the `y` property.

Example 22 Color registration mark settings

In this example, the center of the mark is printed `10.0` millimeters in the horizontal direction and `14.0` millimeters in the vertical direction relative the cursor starting position.

```
<property name="registrationmark">
    <property name="x">10.0,mm</property>
    <property name="y">14.0,mm</property>
</property>
```

Replacing rich black

Rich black is a mixture of cyan, magenta, yellow and black. If rich black is used in an image, you can print the image with a reduced amount of cyan, magenta and yellow ink, or with only black ink. For each pixel in the image, the value of black is compared to the maximum value which is defined in the configuration file. If the black value exceeds or is equal to the maximum value, the amount of cyan, magenta and yellow ink is reduced.

To replace rich black

You configure the maximum value and the print values for cyan, magenta and yellow using the `richblack` property. Specify the:

- Cyan value to print with the `c`.
- Magenta value to print with the `m`.
- Yellow value to print with the `y`.
- Maximum value with the `kmax`.

Example 23 Printing with reduced amount of cyan, magenta and yellow

```
<property name="richblack">2,2,2,254</property>
```

In this example, the following applies:

- A pixel with input values 7,240,8,253 is printed according to the input values.
- A pixel with input values 7,240,8,254 is printed with 2,2,2,254.
- A pixel with input values 7,240,8,255 is printed with 2,2,2,255.

Example 24 Printing rich black with black

```
<property name="richblack">0,0,0,254</property>
```

In this example, the following applies:

- A pixel with input values 7,240,8,253 is printed according to the input values.
- A pixel with input values 7,240,8,254 is printed with black (0,0,0,254).
- A pixel with input values 7,240,8,255 is printed with black (0,0,0,255).

Replacing CMY black

CMY black is a black color achieved by a mixture of cyan, magenta and yellow. If CMY black is used in an image, you can replace it with black ink.

Note: You can only replace CMY black with black if the input values for cyan, magenta and yellow are maximized to a value of 255.

To replace CMY black

You configure the print values for cyan, magenta, yellow and black using the `cmyblack2kblack` property. Specify the:

- Cyan value to print with the `c`.
- Magenta value to print with the `m`.
- Yellow value to print with the `y`.
- Black value to print with the `kmax`.

Example 25 Replace CMY black with black

In this example, a CMY black pixel with input values 255, 255, 255, 70 is printed in black (`0,0,0,255`).

```
<property name="cmyblack2kblack">0,0,0,255</property>
```

Configuring the total ink amount

The total ink amount is set in percentage for all four channels within the range of 0 to 400 percent. You configure the total ink amount with the `inklimit` property in the configuration file.

Example 26 Total ink limit settings

In this case, the total ink amount is set to `400` percent.

```
<property name="inklimit" type="int">400</property>
```

Halftoning

To achieve sufficient picture quality, you can use an error distribution method for halftoning the pictures. The following methods can be used with the IJPDS driver:

- Floyd-Steinberg
- Jarvis
- Stucki

To use halftoning

You configure the error distribution method using the `halftonemethods` property.

Example 27 Error distribution method for halftoning

In this example, the `floyd-steinberg` error distribution method is used for halftoning.

```
<property name="halftonemethods">floyd-steinberg</property>
```

Printheads and RIPs

The printhead is a hardware part that applies ink to paper. A IJPDS printer can have up to eight printheads. If the printer has multiple printheads side by side, normally the first printhead starts printing from the cursor starting position. The second printhead starts printing from a position relative to the cursor starting position plus the width of the first printhead, and so on.

The physical RIP (Raster Image Processor) is the device that controls the printheads, processes input data and builds image to print. The RIP controls a single printhead or multiple stitched printheads. The stitched printheads are treated as a single printhead in the print job, and produce a contiguous image that appears to be produced by one single printhead. A logical RIP is multiple physical RIPs that are stitched.

Configuring printheads

You can configure the number of jets and the number of ink drops to print per dot. You can also configure the relative starting position of the printhead, for example, to compensate for gaps between the printheads.

If all printheads have the same characteristics, you only need to configure one printhead.

The printhead definition must have a unique ID. You use the ID to assign printhead characteristics to a RIP, see *Using a RIP with single printhead* on page 61 and *Example 29* on page 61.

To configure printheads

1. Configure a unique printhead definition ID with the `printhead` property.
2. If required, change the relative starting position for the printhead with the `xposition` property.
3. Configure the number of:
   - Jets with the `jetcount` property.
   - Drops per dot with the `dropcount` property.

Example 28 Printhead configuration

In this example, the printhead uses `2688` jets and prints `1` drop of ink per dot. The position of the printhead is not specified in the job file since `xposition` is `0`.

```
<property name="printheads">
  <property name="printhead" id="0">
    <property name="xposition" type="int">0</property>
    <property name="jetcount" type="int">2688</property>
    <property name="dropcount" type="int">1</property>
  </property>
```

```
</property>
```

Using a RIP with single printhead

To configure the number of jets, the number of drops per dot and a relative starting position of the RIP, you assign a printhead definition to the RIP. For information on how to configure printhead definitions, see *Configuring printheads* on page 60.

To configure which color the RIP should print, you assign a color separation plate to the RIP. For information on how to configure plates, see *Configuring the color a RIP should print* on page 54.

The RIP must be included in a group and each RIP group must have a unique ID. For each group you configure which side of the sheet the RIP group should produce.

Each RIP must also have a unique ID. If you specify a physical RIP, you must use the RIP ID defined in the hardware.

To configure a RIP with a single printhead

1. Configure a unique ID for the RIP group with the `ripgroup` property, for example:

   ```
   <property name="ripgroup" id="0">
   ```

2. Configure which side of the sheet the RIP group should print with the `type` property, for example the front side of the sheet:

   ```
   <property name="type">frontpage</property>
   ```

3. Configure the RIPs included in the RIP group with the `rip` property and give the RIP a unique ID, for example:

   ```
   <property name="rip" id="2">
   ```

4. Assign the color separation plate ID to the RIP with the `plate` property.
5. Assign the printhead definition ID to the RIP with the `printheads` property.

Example 29 RIP settings for monochrome printing with a single printhead

In this example, one RIP with a single printhead prints the front side of the sheet. The printhead prints `black` with `2688` jets and with `1` drop of ink per dot.

The black color is derived from the definition of the color separation plate and the printhead characteristics are derived from the printhead definition.

```
<!-- Color separation plate definition -->
     <property name="plate" id="0">black</property>

<!--Printhead definition -->
<property name="printheads">
   <property name="printhead" id="5">
```

```
      <property name="xposition" type="int">0</property>
      <property name="jetcount" type="int">2688</property>
      <property name="dropcount" type="int">1</property>
   </property>
</property>

<!--RIP group definition-->
<property name="ripgroup" id="0">
   <property name="type">frontpage</property>
   <property name="rip" id="2">
       <property name="plate" type="int">0</property>
       <property name="printheads">5</property>
   </property>
</property>
```

Using RIPs with stitched printheads

If you print with stitched printheads, you must assign a printhead definition for each single printhead. You can use the same definition for all printheads.

You assign the comma separated printhead definition IDs to the RIP with the `printheads` property, for example,
`<property name="printheads">0,1,2,3</property>`

For information on how to configure printhead definitions, see *Configuring printheads* on page 60.

For general information on how to configure RIPs, see *Using a RIP with single printhead* on page 61.

Example 30 A RIP controlling stitched printheads with identical characteristics

In this example, the RIP prints with two printheads. The printheads have the same characteristics, i.e. they both print with `256` jets and `1` drop of ink per dot.

```
<!--Printhead definition -->
<property name="printheads">
   <property name="printhead" id="0">
     <property name="xposition" type="int">0</property>
     <property name="jetcount" type="int">256</property>
     <property name="dropcount" type="int">1</property>
   </property>

<!--RIP group definition-->
<property name="ripgroup" id="0">
   <property name="type">frontpage</property>
```

```
   <property name="rip" id="2">
       <property name="plate" type="int">0</property>
       <property name="printheads">0,0</property>
   </property>
</property>
```

Example 31 A RIP controlling stitched printheads with different characteristics

In this example, the RIP prints with two printheads. Both printheads print `1` drop of ink per dot. The printhead with ID `0` prints with `256` jets and the printhead with ID `1` prints with `1024` jets.

```
<!--Printhead definition -->
<property name="printheads">
   <property name="printhead" id="0">
     <property name="xposition" type="int">0</property>
     <property name="jetcount" type="int">256</property>
     <property name="dropcount" type="int">1</property>
   </property>
<property name="printhead" id="1">
     <property name="xposition" type="int">0</property>
     <property name="jetcount" type="int">1024</property>
     <property name="dropcount" type="int">1</property>
   </property>
</property>

<!--RIP group definition-->
<property name="ripgroup" id="0">
   <property name="type">frontpage</property>
   <property name="rip" id="2">
       <property name="plate" type="int">0</property>
       <property name="printheads">0,1</property>
   </property>
</property>
```

Using stitched RIPs

Multiple physical RIPs can be stitched to form a logical RIP. For example, two stitched physical RIPs with one printhead each form one logical RIP with two printheads.

The printer operator must manually configure the system controller and data station when you use stitched RIPs in a job.

For general information on how to configure RIPs, see *Using a RIP with single printhead* on page 61.

To configure stitched RIPs

You configure which physical RIPs to stitch using the `stitching` property. For example, `<property name="stitching">0,1</property>`

Example 32 Stitched RIPs

In this example two physical RIPs are stitched to form a logical RIP that prints the front side of the sheet. The physical RIPs have ID `0` and `1`, which are defined in the hardware.

The RIP with ID `0` prints with `2688` jets and prints `1` drop of ink per dot. The RIP with ID `1` uses the same number of jets, but prints `2` drops of ink per dot.

```
<!--Printhead definition -->
<property name="printheads">
   <property name="printhead" id="2">
     <property name="xposition" type="int">0</property>
     <property name="jetcount" type="int">2688</property>
     <property name="dropcount" type="int">1</property>
   </property>
   <property name="printhead" id="3">
     <property name="xposition" type="int">0</property>
     <property name="jetcount" type="int">2688</property>
     <property name="dropcount" type="int">2</property>
   </property>
</property>

<!--RIP group description-->
<property name="ripgroup" id="0">
   <property name="type">frontpage</property>
   <property name="rip" id="1">
     <property name="plate" type="int">0</property>
     <property name="printheads">2,3</property>
     <property name="stitching">0,1</property>
   </property>
</property>
```

Printing cue marks

You can print cue marks on the web. The printing system uses the cue mark to synchronize the start of a document.

A cue mark is normally a 4 by 4 millimeter black square. The mark is printed in a position relative the cursor starting position (the upper left corner of the IJPDS document).

To print cue marks

You configure the size and position of the cue mark using the `cuemark` property. You configure the cue mark:

- Width with the `w` property.
- Height with the `h` property.
- Upper left corner horizontal position with the `x` property.
- Upper left corner vertical position with the `y` property.

Example 33 Cue mark settings

In this example, a 4 by 4 millimeters cue mark is printed. The upper left corner of the mark is printed `11.0` millimeters in the horizontal direction from cursor starting position.

```
<property name="cuemark">
   <property name="x">11.0,mm</property>
   <property name="y">0.0,mm</property>
   <property name="w">4.0,mm</property>
   <property name="h">4.0,mm</property>
</property>
```

Generating a RIP setup file

You can generate a RIP setup file and use this file to set up the RIP positions, colors and web sides in the IJPDS proofer software. The RIP setup file has the same name as the job file, but with the extension `ijq`.

You must use a File output connector to generate the RIP setup file.

To generate a RIP setup file

You use the `generateijq` property to specify whether to generate a RIP set up file.

Example 34 RIP setup file settings

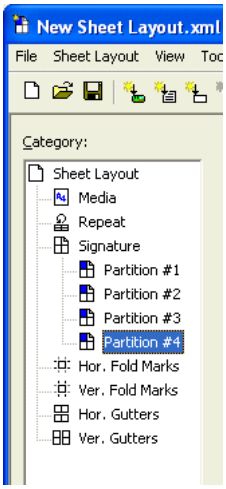
In this case, a RIP setup file is generated.

```
<property name="generateijq" type="bool">yes</property>
```

Duplex and n-up printing

For duplex printing you must configure one RIP group for each side of the sheet. Each RIP group can comprise of one or multiple RIPs.

If you print multiple document pages on the same side of the sheet (n-up printing) or if you configure more than two RIP groups, you must use a sheet layout. The sheet layout must be divided into partitions. You use the partition number to configure which RIP should print a specific part of the sheet. The partition numbers are displayed in the Sheet Layout editor in Design Center.



For information on how to create and use sheet layouts in a Project, see the Sheet Layout documentation.

To configure duplex printing

You configure which side of the sheet the RIP group should print using the `type` property, for example `<property name="type">frontpage</property>`

Example 35 RIP settings for duplex printing

In this example, the front side of the sheet is printed in color by four RIPs (ID `0`, `1`, `2` and `3`). The back side of the sheet is printed by one RIP (ID `4`) with black. All the RIPs have two stitched printheads.

```
<!-- RIP group 0 for front page -->
<property name="ripgroup" id="0">
   <property name="type">frontpage</property>
   <property name="rip" id="0">
     <property name="plate" type="int">0</property>
     <property name="printheads">0,0</property>
   </property>
   <property name="rip" id="1">
     <property name="plate" type="int">1</property>
     <property name="printheads">0,0</property>
```

```
   </property>
   <property name="rip" id="2">
     <property name="plate" type="int">2</property>
     <property name="printheads">0,0</property>
   </property>
   <property name="rip" id="3">
     <property name="plate" type="int">3</property>
     <property name="printheads">0,0</property>
   </property>
</property>

<!-- RIP group 1 for back page -->
<property name="ripgroup" id="1">
   <property name="type">backpage</property>
   <property name="rip" id="4">
     <property name="plate" type="int">3</property>
     <property name="printheads">0,0</property>
   </property>
</property>
```

To configure n-up printing

You configure the sheet layout partition number using the `partition` property.

Example 36 RIP settings for n-up printing

In this example, the RIP prints the part of the sheet front side that is defined as partition `1` in the sheet layout configuration.

```
<property name="ripgroup" id="0">
   <property name="type">frontpage</property>
<!--Sheet partition number used to map sheet partition to RIP -->
     <property name="partition">1</property>
     <property name="rip" id="0">
     <property name="plate" type="int">0</property>
     <property name="printheads">0</property>
   </property>
</property>
```

Converting overlays to fixed files

Each time an LXF or TIFF overlay is used, it is converted to a fixed file and included in the data stream. To improve job performance you can specify that overlays should be converted and included in the job file and referenced in the data stream.

If not enough memory is allocated in the printer for fixed files, the overlays are converted each time they are used.

To convert overlays to fixed files

You convert the overlays to fixed files using the `overlays2fixedfiles` property.

Example 37 Conversion of overlays to fixed files

In this case, the overlays are included once in the job file and referenced in the data stream.

```
<property name="overlays2fixedfiles" type="bool">yes</property>
```

To allocate memory for fixed files

You allocate memory for fixed files using the `fixedfilememory` property.

Example 38 Allocation of memory for fixed files

In this case, `5242880` bytes are allocated for fixed files.

```
<property name="fixedfilememory" type="int">5242880</property>
```

Compressing bitmap

You can compress bitmaps to reduce the memory required for bitmap images and font characters.

To compress bitmaps

You use the `compression` property to specify whether to compress bitmaps or not.

Example 39 Bitmap compression settings

In this case, the bitmaps are compressed.

```
<property name="compression" type="bool">yes</property>
```

Allocating resource memory

The resource memory is mainly used for fonts. You must specify a size for the resource memory that corresponds to the actual size of the memory available in the printer.

To allocate resource memory

You allocate resource memory using the `resourcememory` property.

Example 40 Settings for resources memory size

In this example, `266338304` bytes are allocated for resource memory.

```
<property name="resourcememory" type="int">266338304</property>
```

Splitting large jobs

Jobs that exceed two gigabytes are split into multiple output files by the Kodak Versamark IJPDS printing system. A sequence number is appended to the name of each output file.

If your printing system does not support jobs larger than two gigabyte, you can configure a maximum job size. The job is split before the maximum size is reached, i.e. the output files will not be exactly the size of the maximum size. The split is made at the closest allowed offset and only at IJPDS document level.

You must use a File output connector to split jobs.

To split a large job

You specify at which size you want to split the job using the `filesplitthreshold` property.

Example 41 Splitting large job settings

In this case, the job is split before one gigabyte is reached.

```
<property name="filesplitthreshold" type="int">1000000000
</property>
```

IJPDS driver quick reference

Resolution	
x-resolution	Horizontal resolution in DPI.
y-resolution	Vertical resolution in DPI.

Color manager	
richblack	Values are set for `c`, `m`, `y` and `kmax`. If K black in images are equal to or larger than `kmax`, the amounts of cyan, magenta and yellow ink are reduced.
cmyblack2kblack	Values are set for `c`, `m`, `y` and `kmax`. CMY black in images are replaced by black, using `kmax` value.
inklimit	Total maximum ink limit for all four channels, given in a percentage within the range `0` to `400`.
rgb2cmykprofile	ICC color matching profile file used to convert RGB images to CMYK.
renderingintent	Rendering intent used with the ICC color matching profile. Available values are `picture`, `graphic`, `proof` and `match`.
lut	Unique ID for a set of color lookup tables.
lutc	Color lookup tables for cyan.
lutm	Color lookup tables for magenta.
luty	Color lookup tables for yellow.
lutk	Color lookup tables for black.
imagetable	Name of the image lookup table file.
imagelut	ID of the color lookup table set used for images.
vectorlut	ID of the color lookup table set used for vector graphics.
textlut	ID of the color lookup table set used for text.

Halftoning and color separation	
halftonemethods	Error distribution method to use for halftoning. Available values are `floyd-steinberg`, `jarvis` and `stucki`.

Halftoning and color separation	
plate	Halftoning plate. Available values are `cyan`, `magenta`, `black` and `yellow`.

Marks	
registrationmark	Registration mark for color separation. `x` – Horizontal position of the registration mark center, relative to the cursor starting position in millimeters. `y` – Vertical position of the registration mark center, relative to the cursor starting position in millimeters.
cuemark	Cue mark representing the start of a IJPDS document. `x` – Horizontal position of the cue mark upper left corner, relative to the cursor starting position in millimeters. `y` – Vertical position of the cue mark upper left corner, relative to the cursor starting position in millimeters. `w` – Cue mark width in millimeters. `h` – Cue mark height in millimeters.

Overlays	
overlays2fixedfiles	`yes` – Convert LXF overlays into fixed files and include them in the job file. `no` – Convert and include LXF overlays each time they are used.

Bitmap compression	
compression	`yes` – Compress bitmaps and font characters using VRLE32 compression. `no` – Compression is not used.

RIP setup file	
generateijq	`yes` – Generate a RIP setup file (`*.ijq`) for the Kodak proofer. `no` – No RIP setup file is generated.

Memory allocation	
resourcememory	Available memory in bytes for resources.
fixedfilememory	Available memory in bytes for fixed files.

Job split	
filesplitthreshold	The size in bytes at which large jobs are split into multiple output files.

Printhead definitions	
printhead	Unique ID of the printhead definition.
xposition	The offset of the printhead relative to the starting position of the first printhead. This is specified in the number of dots.
jetcount	Number of jets in the printhead.
dropcount	Number of ink drops per dot.

RIP groups	
ripgroup	Unique ID for a group of RIPs.
type	RIP group type, which defines on which side of the sheet the printheads are positioned. Possible values are `frontpage` and `backpage`.
partition	Sheet layout partition number.
rip	ID of the physical RIP, or logical RIP if multiple physical RIPs are stitched.

RIP groups	
plate	Color separation plate used by the RIP.
printheads	Comma-separated list of printhead definitions used by the RIP.
stitching	Comma-separated list of physical RIPs that are stitched.

PCL 6

The PCL 6 driver is a Hewlett-Packard developed LaserJet PCL language with related technology.

User interface

You configure the PCL 6 driver in the Output Connector Settings and the Runtime Output Connector Settings dialog boxes.

Output Connector Settings

The following settings can be made in the Design Center Output Connector Settings dialog box.

Page size

Specifies the size of the target page.

Value	Description
Letter	Letter paper size.
Legal	Legal paper size.
A4	A4 paper size.
Executive	Executive paper size.
Ledger	Ledger paper size.
A3	A3 paper size.
Comm-10 Env	COM10 envelope size.
Monarch Env	Monarch envelope size.
Intl. C5 Env	C5 envelope size (International).
Intl. DL Env	DL envelope size (International).
JIS B4	JB4 paper size (Japan Industrial Standard).
JIS B5	JB5 paper size (Japan Industrial Standard).
Intl. B5 Env	B5 envelope size (International).
J Postcard	J postcard size.
J Double Postcard	J double postcard size.
A5	A5 paper size.

Value	Description
A6	A6 paper size.
JIS B6	JB6 paper size (Japan Industrial Standard).
Custom Size (mm)	Custom page size in millimeters. The size is defined by the `CustomPageWidth` and `CustomPageHeight` options.

CustomPageWidth

The width (mm) of a custom media size. This option is active when the Page Size value is set to `Custom Size`.

CustomPageHeight

The height (mm) of a custom media size. This option is active when the Page Size value is set to `Custom Size`.

Paper source

The printer source from which the media will be chosen.

Value	Description
Auto Select	The media source is automatically selected.
Manual Feed	The media in the manual feed source.
MultiPurpose Tray	The media in the multipurpose tray.
Upper Cassette	The media in the upper media cassette.
Lower Cassette	The media in the lower cassette.
Envelope Cassette	The media in the envelope tray.
Third Cassette	The third cassette (only on printers equipped with this feature).
External Tray 1	The media in the first external input tray.
External Tray 2	The media in the second external input tray.
External Tray 248	The media in the 248th external input tray.

PageDestination

The destination for each page after it has been printed.

Value	Description
Default Destination	This is the bin currently set by the operator.
Face Down Bin	Sets the paper output to the facedown bin.
Face Up Bin	Sets the paper output to the face up output bin (only on printers equipped with this feature).
Job Offset Bin	Sets the paper output to the job offset bin (only on printers equipped with this feature).
External Bin 1	The media in the first external output bin.
External Bin 2	The media in the second external output bin.
External Bin 251	The media in the 251th external output bin.

Duplex

The pages will be printed on either one side or on both sides of the media, and oriented for either horizontal or vertical binding.

Value	Description
None	Using one page per physical media page.
Horizontal	Using a horizontal binding of the printed pages.
Vertical	Using a vertical binding of the printed pages.

DuplexSide

This option specifies the side of the media on which the current page should be printed.

Value	Description
None	The device default setting.
Front Media Side	The front side of the physical pages.
Back Media Side	The back side of the physical pages.

Orientation

The orientation of the page.

Value	Description
Portrait	Portrait page orientation.
Landscape	Landscape page orientation.
Reverse Portrait	Reverse portrait page orientation.
Reverse Landscape	Reverse landscape page orientation.

ImageColorQuality

The color quality of the output image.

Value	Description
256 Color	256 color quality (8 bit).
High Color	Maximum color quality (24 bit).

ImageCompression

The compression method used on the output image.

Value	Description
No Compression	No compression.
RLE Compression	Run length encoding (RLE) compression.

Runtime Output Connector Settings

The following settings can be made in the Design Center Runtime Output Connector Settings dialog box.

ByteOrder

The byte order specifies the ordering of the most significant to the least significant bytes in the stream for multibyte binary fields.

Choice name	Description
Big Endian	A binary binding follows in the stream body where operator identifiers, attribute identifiers, and attribute values are expressed in a form where the most significant byte is the first byte in the binary field (from left to right) and the least significant byte is last (to the right).
Little Endian	A binary binding follows in the stream body where operator identifiers, attribute identifiers, and attribute values are expressed in a form where the least significant byte is the first byte in the binary field (from left to right) and the most significant byte is last (to the right).

ErrorReport

Indicates the method for how errors are reported to the user.

Choice name	Description
Error Page	Errors will be reported by printing or displaying an error page.
No Reporting	No error reporting will be performed.
Back Channel	Errors will be reported through the back channel (i.e. device dependent).
Back Channel and Error Page	Errors will be reported through the back channel and by printing or displaying an error page.
No Warn. Back Channel	Only errors will be reported (no warnings) through the back channel (i.e. device dependent).
No Warn. Error Page	Only errors will be reported (no warnings), only by printing or displaying an error page.
No Warn. Back Channel and Error Page	Only errors will be reported (no warnings) through the back channel and by printing or displaying an error page.

PDF

The following applies to the PDF driver:

- Generates the lowest possible PDF version, which is normally PDF 1.3. If the output contains features that require PDF 1.4, then PDF 1.4 is generated.
- Compatible with Acrobat 4.0 and later (see *Acrobat - PDF compatibility* on page 83).
- Accepts LXF overlays.
- Enables adding and embedding of TrueType fonts in output.
- Enables adding and embedding of Postscript Type 1 fonts in output.
- Full Unicode support (via automatic font embedding).
- Full color support.
- Accepts graphics added in PageOUT.
- Supports rounded corners and oblique lines.
- Supports file compression.
- Supports encryption.
- Supports bookmarks.

To generate PDF output, you must select and configure a PDF driver for the output connector, and configure the corresponding driver options, see *PDF driver settings* on page 85.

Fonts

You can use TrueType or Postscript Type 1 fonts in the PDF output.

Embedding fonts

Embedded TrueType or Postscript Type 1 fonts enable the receiver of the PDF file to view and print fonts without installing them locally. For information on how to embed fonts in PDF documents, see *Embedding fonts in PDF documents* on page 121.

Acrobat - PDF compatibility

Acrobat version	Native file format (PDF version)
Acrobat 3.0	PDF 1.2
Acrobat 4.0	PDF 1.3
Acrobat 5.0	PDF 1.4

References

For more information about Acrobat and PDF, go to
http://partners.adobe.com/asn/techresources.jsp

PDF driver settings

Setting	Description
Use compression	• **Yes** – Use compression. • **No** – Do not use compression. Use compression whenever possible.
Info: Author	The document author.
Info: Creator	The application that created the document.
Info: Title	The document title.
Info: Subject	The document subject.
Info: Keywords	Keywords associated with the document.
Use encryption	• **No** – Do not use encryption. • **Standard R2** – Use Standard Security Handler revision 2 (encryption algorithm RC4 with 40-bit keys). See *Standard Security Handler revision 2* on page 90. • **Standard R3** – Use Standard Security Handler revision 3 (encryption algorithm RC4 with 128-bit RC4 encryption). See *Standard Security Handler revision 3* on page 90.
User password	***Applicable only if encryption is used*** A password a user must enter to open, decrypt, and display the document on screen. Enables printing, modifying, and copying of the document according to the values selected for the Allow print, Allow modify, and Allow copy options below.
Owner password	***Applicable only if encryption is used*** A password a user must enter to open, decrypt, and display the document on screen. Enables full access to the document.
Allow print	***Applicable only if encryption is used*** • **No** – Do not allow printing of the document. • **Yes** – Allow printing of the document.
Allow modify	***Applicable only if encryption is used*** • **No** – Do not allow modification of the document. • **Yes** – Allow modification of the document.

Setting	Description
Allow copy	***Applicable only if encryption is used*** • **No** – Do not allow copying of text and graphics from the document. • **Yes** – Allow copying of text and graphics from the document.
Use filter ASCII 85	• **Yes** – Encode data as 7-bit ASCII text. • **No** – Do not encode data as 7-bit ASCII text. The ASCII-85 filter encodes binary data as 7-bit ASCII text consisting only of printable characters. This enables data to be sent through various communication channels without undesired changes, for example, older mail systems that cannot handle 8-bit data streams. When this filter is used, the overall PDF file size becomes approximately 20% larger. Encrypted PDF files are always binary, so the filter is not useful for this kind of data. For more information about the ASCII-85 filter, see for example the PostScript® Language Reference.
Use resolution	• **No** – Embed images as they are. • **Yes** – The resolution specified in the driver file is used. The default resolution is 300 dpi. Some images are downscaled, depending on the actual image resolution and the resolution specified in the driver file. To specify another value than the default resolution you must edit the driver file. See the *Device driver tools* documentation.
Embed threshold	If you want to embed characters with codes below 256, you can define which characters to embed when using Unicode characters in TrueType fonts. For example, if you set this value to 128, all characters used, with codes greater than or equal to 128 will be embedded. Minimum value is 0, maximum value is 256.
Viewer: Page number	Applied when the user opens the document. The first page displayed in the viewer.
Viewer: Page zoom	Applied when the user opens the document. • **Default** – Use the display settings in Acrobat general preferences. • **Fit in window** – Adjust the page size to fit the window size. • **Fit width** – Adjust the page size to fit the window width. • **Fit visible** – Adjust the page size to fit the window width, displaying all visible text and graphics but not the page margins. • **25%-1600%** – The zoom factor.

Setting	Description
Viewer: Page layout	Applied when the user scrolls the document. • **Default** – Use the display settings in Acrobat general preferences. • **Single page** – Display just one page at the time. • **Continuous** – Display pages continuously. • **Continuous Facing** – Display even and odd pages facing each other and continuously.
Viewer: Full screen	Applied when the user opens the document. • **No** – Do not open the document in full screen view. • **Yes** – Open the document in full screen view.
Viewer: Hide toolbar	Applied when the user opens the document. • **No** – Do not hide the toolbar • **Yes** – Hide the toolbar.
Viewer: Hide menu bar	Applied when the user opens the document. • **No** – Do not hide the menu. • **Yes** – Hide the menu bar.
Viewer: Hide window UI	Applied when the user opens the document. • **No** – Do not hide the window controls. • **Yes** – Hide the window controls, for example bookmarks and thumbnails.
Viewer: Fit window	Applied when the user opens the document. • **No** – Do not resize the viewer to fit the page. • **Yes** – Resize the viewer to fit the page.
Viewer: Center window	Applied when the user opens the document. Select **Yes** to display the window centered on screen.
Viewer: Display document title	Applied when the user opens the document. • **No** – Do not display the document title in the title bar. • **Yes** – Display the document title in the title bar.
Viewer: Page mode	Applied when the user opens the document. • **Page Only** – Only display the page. • **Bookmarks and Page** – Display the Bookmarks tab and the page.

Setting	Description
Conformance	The conformance level of the PDF output. **Plain PDF** Select to generate plain PDF. **Tagged PDF** Select to generate tagged PDF, i.e. include information about the structure of the document in the PDF. Structure is the term for a set of instructions that define the logic that binds the content together – for example the correct reading order, and the presence and meaning of significant elements such as figures, lists, tables, etc. **Note:** Tagged PDF can only be used in combination with the Adobe LiveCycle Designer ES Process. **PDF/A-1a** Select to generate PDF/A Level A compliant output. See *About PDF/A* below for more information. **Note:** PDF/A-1a can only be used in combination with the Adobe LiveCycle Designer ES Process. **PDF/A-1b** Select to generate PDF/A Level B compliant output. See *About PDF/A* below for more information. **About PDF/A** PDF/A is used for long term archiving of electronic documents, and ensures the documents can be reproduced the exact same way in years to come. All information necessary for displaying the document is embedded in the file. This includes all content, fonts, and color information. A PDF/A document must not be reliant on information from external sources (e.g. font programs and hyperlinks), and must not be password protected. PDF/A Level B is focused on preserving the appearance of the document. PDF/A Level A is the same as PDF/A Level B, with the addition of tagged PDF. Level A conformance ensures, in addition to the appearance of the document, that the document structure is described in the PDF, and that all text can be extracted. For example, to enable a document to be read out loud using a speech synthesizer, you must use Level A.

Setting	Description
Use JPEG Image Compression	Specifies whether to convert color images to compressed JPEG images. • **Yes** – Convert all color image formats to compressed JPEG images. Monochrome images will be compressed as CCIT G4 or using ZLIB. • **No** – Keep the original image formats. No compression is applied unless the original format is JPEG. **Comments** If you select **No**, JPEG compression will still be used for images already in JPEG format. If a source image is of any other format (PNG, BMP, etc.) you must select **Yes** to compress the image in the output document. Note that if you select **No**, and the option Use compression above is set to **Yes**, the ZLIB compression is applied to images.
JPEG Image Compression Quality	Specifies the compression level and quality of JPEG images in the output document. You can select a number between 10 (highest compression level/worst quality) and 100 (lowest compression level/ best quality). The default value is 80. **Comments** Compression quality is applied when compressing non-JPEG images. Images already in JPEG format are not reprocessed.
Copies	Not applicable.

Encryption

You can encrypt the PDF output to protect the generated PDF documents from unauthorized access. Encryption only applies to the contents of the document. It does not apply to other object types that are used to convey information about the structure of the document.

The filter that controls access to the contents of the encrypted document is the security handler. The security handlers available in PDF are Standard Security Handler revision 2 and Standard Security Handler revision 3.

Standard Security Handler revision 2

Algorithm RC4 with 40-bit keys.

Passwords User and owner passwords can be specified.

Standard Security Handler revision 3

Limitations Acrobat version 4.05 and PDF version 1.4 or higher.

Algorithm RC4 with 128-bit keys.

Passwords User and owner passwords can be specified.

PDF (PCL Convert)

The following applies to the PDF (PCL Convert) driver:

- Generates PDF version 1.2.
- Compatible with Acrobat 3.0 (see *Acrobat - PDF compatibility* on page 91).
- Accepts PCL overlays.
- Enables inclusion of soft fonts in output.
- No color support.
- Accepts graphics added in PageOUT.
- Supports rounded corners and oblique lines (only if added to PageOUT as overlays).
- Does not support file compression.
- Does not support encryption.

To generate PDF (PCL Convert) output, you must select and configure a PDF driver for the output connector, and configure the appropriate driver options, see *PDF (PCL Convert) driver settings* on page 92

Soft fonts in PDF output

You can include soft fonts in PDF (PCL Convert) output. This enables the receiver of the document to view and print fonts that are not locally installed. For information on how to download soft fonts to PCL printers, see *Downloading soft fonts to PCL printers* on page 122.

Acrobat - PDF compatibility

Acrobat version	Native file format (PDF version)
Acrobat 3.0	PDF 1.2
Acrobat 4.0	PDF 1.3
Acrobat 5.0	PDF 1.4

References

For more information about Acrobat and PDF, go to

http://partners.adobe.com/asn/techresources.jsp

PDF (PCL Convert) driver settings

PDF (PCL Convert) settings	
Page Size	Page size.
Print Orientation	Print orientation.
Copies	N/A

PostScript

The StreamServe PostScript driver supports PostScript Level 2.

Image compression

Images are compressed using:

- CCITT compression for 1-bit monochrome images
- LZW (Lempel-Ziv-Welch) compression for grayscale and color images (i.e. all but 1-bit monochrome images).

The appropriate compression method is selected automatically according to image type. To further optimize image output for non-color printers, set the **Grayscale** option to **Yes**.

PostScript forms

PostScript forms can be used to optimize driver output. A form is a resource that can be added to the output stream once, cached in the printer memory and then executed multiple times on subsequent pages. This means that only information that changes between forms needs to be interpreted for each page. Images (raster and EPS), device overlays (EPS), and LXF overlays can be converted into forms.

Note: Dynamic overlays from PreformatIN can not be converted into forms since they only appear once in the printed file.

Settings	
Page size	Page size. If you select **None**, the printer device settings will be used when printing. **Note:** If you use a sheet layout definition, the sheet layout settings will override the driver settings.
Simplex/Duplex	Select whether to print on one side or both sides of the paper. If you select **None**, the printer device settings will be used when printing. **Note:** If you use a sheet layout definition, the sheet layout settings will override the driver settings.
Orientation	Print orientation. If you select **None**, the printer device settings will be used when printing. **Note:** If you use a sheet layout definition, the sheet layout settings will override the driver settings.
For	The name of the user for whom the document is printed.
Routing	Information about how to route the document back to its owner after printing, for example an email address.

Settings	
Version	The version or revision number of the document or resource.
Page order	**None** – Select this option if the order of the pages is not important. **Ascend** – The pages of the document are in ascending order. **Descend** – The pages of the document are in descending order. **Special** – The pages in the document are in a special order, e.g. signature order and should not be changed.
Grayscale	**Yes** – Converts color images into 8-bit grayscale and compresses them using the LZW compression method. Select this option to optimize image output for non-color printers. **No** – The compression method is selected automatically.
Use PostScript forms	**Yes** – All static images and overlays used in the Project are generated as PostScript form objects. **No** – All data is processed as usual.
Virtual Memory for Forms (KB):	The amount of virtual memory to be used for forms. The default form cache size is 16 MB (16384 kB). If the amount of cached PostScript data exceeds the limit defined for this setting, the image or overlay will not be cached and will be processed as usual.
Copies	Not applicable.

XPS

The XPS driver produces Microsoft XPS documents according to XML Paper Specification version 1.0.

XPS (XML Paper Specification) is a document storage and viewing specification developed by Microsoft. It is based on the XML language and the printer page description language is based on:

- a new print path.
- a color-managed device independent and a resolution independent vector-based document format.
- support for advanced printing features such as:
  - gradients
  - transparencies
  - CMYK color space
  - named colors
  - printer calibration
  - print schemas etc.

The XPS document format contains a set of related pages with a fixed layout organized as one or more documents. A file that implements this format includes everything necessary to fully render these documents on a display device or physical medium (e.g. paper). This includes all resources such as fonts and images that might be required to render the individual page markings. XPS uses a ZIP archive, containing documents (fixed documents, pages etc.) and resources (images, fonts etc.).

The XPS documents use the XML language to describe fixed documents and pages. The documents contain a root fixed document sequence that binds a collection of fixed documents which, in turn, bind a collection of fixed pages.

For more information about the XPS driver, see the XPS website on `www.microsoft.com`.

Copyright

For use of the StreamServer XPS driver, the following copyrights apply:

Copyright © 1990-2005 Info-ZIP. All rights reserved.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

Redistributions in binary form (compiled executables) must reproduce the above copyright notice, definition, disclaimer, and this list of conditions in documentation and/or other materials provided with the distribution.

Zebra Label Printer

The Zebra Label Printer driver automatically generates ZPLII (Zebra Programming Language II) code. ZPLII is a high-level label definition and printer control language. It contains instructions for printer configuration, label text, bar codes, lines, and bitmap graphics.

For information and details about ZPLII, see the *ZPLII Programming Guide*, Zebra Part Number 46496L, published by Zebra Technologies Corporation. This document is available at http://www.zebra.com/SS/manuals.htm

Required knowledge

This manual is intended for users who are familiar with StreamServe and ZPLII.

Upgrading existing Projects using the Zebra Label Printer driver

Projects that use the old version of the driver (Zebra Label Printer StreamServe 3.x), cannot be automatically upgraded to use the new driver.

If possible, you should consider replacing the old driver with the new Zebra Label Printer driver, which has improved functionality and is easier to use, especially when working with barcodes and graphics.

However, if required, you can continue to use the Zebra Label Printer driver for StreamServe 3.x in StreamServe 4.x.

Driver files and printer resolution

You should use the Zebra Label Printer driver that corresponds to the resolution supported by your Zebra printer:

- ZEBRA LabelPrinter (150dpi)
- ZEBRA LabelPrinter (200dpi)
- ZEBRA LabelPrinter (300dpi)
- ZEBRA LabelPrinter (600dpi)

Getting the label size right

The Zebra Label Printer driver supports any label size supported by the printer.

You define the label size in PageOUT. When printed, the label is automatically centered on the media. The best result is achieved if the page width defined in PageOUT is equal to the actual physical width of the media.

In cases where the page size is larger than the physical size of the label, the output will be different from that defined in PageOUT or the overlay.

Continuous media

When using continuous media for printing, the label length is defined by the page height.

Overlays

If you use overlays, the size of the overlay should be the exact same size as that of the page defined in PageOUT.

Fonts

The Zebra Label Printer driver supports the A-H bitmap fonts, and the scalable font 0, that come with most Zebra printers. The driver also supports downloading TrueType fonts to the printer.

Limitations

- The GS (Graphic Symbol) Zebra font is supported for all fonts and code pages, except for downloaded Unocide fonts.
- You cannot download Windows TrueType fonts as Zebra bitmap fonts.
- Font rasterization is not supported. Adding the options `Rasterizefont` or `FontEmbed` in the driver file will have no effect.

Tips and caveats

- You should map monospaced TrueType fonts to Zebra fonts A-H, and variable width fonts to the Zebra font 0.
- Zebra fonts A-H are only scalable to specific font sizes. The height of these fonts can differ significantly from what you defined in PageOUT or in the overlay. See *Font sizes* on page 102
- Avoid using horizontal text stretching and scaling for Zebra fonts A-H. This does not work for these bitmap fonts.
- Letter spacing will be ignored for fonts mapped to Zebra fonts A-H.
- Downloading TrueType fonts to the printer will significantly delay the output of the first label in a Job. Using Zebra fonts A-H will speed up processing in the printer.
- Fonts that are downloaded to the printer are only available during the StreamServe Job. The fonts are deleted from the printer memory when the Job is finished.
- Text can only be rotated 0, 90, 180, and 270 degrees. Any other angle will result in 0 degrees rotation in the output.
- Zebra font H only supports uppercase letters. No lowercase letters will be included in the output.

Mapping fonts to Zebra fonts

To change the mapping of Zebra fonts, you must edit the `Fonts Mapping` section in the driver file. See the *Device driver tools* documentation for information on how to edit driver files.

Text underlining and striking out is supported for all Zebra fonts. You can map several fonts to the same Zebra font.

Example 42 Mapping fonts used in PageOUT to Zebra fonts

To map the Arial font to the Zebra font 0, add the following line to the `Fonts Mapping` section in the driver file:

```
MapFont "Arial"                "0"
```

Example 43 Mapping underlined fonts

To map underlined fonts, add a line in the following format to the `Fonts Mapping` section in the driver file:

```
MapFont "Arial_Underline"      "0_Underline"
```

Font sizes

Zebra fonts A-H are bitmap fonts with a fixed width and height. Font sizes specified in PageOUT, or in overlays, will be matched to the nearest Zebra font size, as specified in the driver file.

To change the size of Zebra fonts used in the printed output, you must edit the `Fixed size fonts` section in the driver file. See the *Device driver tools* documentation for information on how to edit driver files.

All font sizes used in PageOUT, and in overlays, must be mapped to a valid Zebra font size. Valid Zebra font sizes for bitmap fonts A-H are multiples of height from 2 to 0 times the standard height in increments of 1. For example, Zebra font A has standard height 9 dots. Valid values for this font would be 9 (default), 18, 36, 45, 54, 63, 72, 81, and 90. Specifying another value will not generate an error, but has no sense because it will be rounded to the closest valid value. See the *ZPLII Programming Guide* for valid font sizes for specific printer resolutions.

Note: In PageOUT, and in overlays, font size is specified in points, while the size of Zebra fonts is specified in printer dots.

Example 44 Mapping font sizes used in PageOUT to Zebra font sizes

To map sizes 10, 12, and 15 of the Courier font to sizes 18, 36, and 45 of Zebra font A, add the following lines to the `Fixed size fonts` section in the driver file:

```
Font "Courier" WidthTable 2
Size 10 SelectPrefix "A"       SelectPostfix "18"
Size 12 SelectPrefix "A"       SelectPostfix "36"
Size 15 SelectPrefix "A"       SelectPostfix "45"
```

Where:

- the `WidthTable` number must correspond to the width table of the selected Zebra font, as specified in the section `Mapped fonts definition` in the driver file. See *Valid width table numbers for Zebra fonts* on page 103 for valid width table numbers.

- the size (10, 12, and 15 in this example) is the size of the font used in PageOUT, specified in **points**.
- the string in double quotes after `SelectPrefix` (A in this example) is the Zebra font.
- the string in double quotes after `SelectPostfix` (25, 35, and 45 in this example) is the Zebra font size (height), specified in **dots**.

Valid width table numbers for Zebra fonts

0	1
A	2
B	3
C	4
D	4
E	5
F	6
G	7
H	8

Downloading TrueType and Unicode fonts to the Zebra printer

To ensure that the printed output looks exactly as defined in PageOUT and overlays, you can download TrueType fonts directly to the printer memory. If you use international character sets, you must download the Unicode fonts to the printer. The downloaded fonts are only available during the StreamServe Job and are deleted from the printer memory when the Job is finished.

To download fonts to the Zebra printer you must do the following:

- Download and install the Zebra Tools Utility

  Download the Zebra Tools Utility (ZTools) from http://www.zebra.com/SD/utilities.htm
- Configure the Zebra printer

  You must convert TrueType fonts, and Unicode fonts if you use international character sets, to the internal ZPLII format. In ZTools, you specify the source directories where TrueType and Unicode fonts are stored, and the path to the destination files. See *Configuring the Zebra printer* on page 104.

- Edit the driver file and add the fonts to be downloaded to the printer
  See *Editing the driver file and adding the fonts to be downloaded to the printer* on page 105.
- Make sure that the TrueType and Unicode fonts, as well as the Zebra fonts, are stored in the specified directories

---

Downloading ZST and ZSU files directly to the printer

If you are familiar with ZPLII and ZTools, you can use ZTools (or other label printer utilities) to download ZPL code, including ZST and ZSU files, to the printer.

Edit the ZST or ZSU file by using the ^CW command to assign a built-in font between I and Z to the font in the file, and download the ZST or ZSU file offline. See the *ZPLII Programming Guide*.

Add the font to the Zebra Label Printer driver file by entering a line in the following format to the `Downloadable Fonts` section:

```
Font "SAMPLE"
    ReadFont "SAMPLE.TTF" Select "Z:ZSU"
```

For example:

```
Font "Arial"
    ReadFont "arial.ttf" Select "Z:ZSU"
```

This will significantly speed up processing in the printer.

Configuring the Zebra printer

You use ZTools to

- specify the directory where TrueType and Unicode fonts are stored, and the destination directory to which Zebra fonts will be written
- convert TrueType fonts to the internal ZPLII format
- convert Unicode fonts to the internal ZPLII format.

To specify the fonts and destination directories

1. In ZTools, select **Options** > **Directories**. The Directories dialog box opens.
2. Specify the path to the font directory.
3. Set the Destination path to the font directory of the current Project.
4. Click **OK**.

To convert TrueType fonts to internal ZPLII format

1. In ZTools, select **Convert** > **TrueType**. The Convert TrueType dialog box opens.

2 Browse to the TrueType font **Source** directory.

3 Click **Convert**.

You should not edit the Destination field. Zebra fonts will be written to the Destination directory you specified in the Directory configuration. The destination files have the extension ZST.

Note: Do not edit the ZST file.

To convert Unicode fonts to internal ZPLII format

1 In ZTools, select **Convert** > **Unbound TT**. The Convert Unbound TrueType dialog box opens.

2 Browse to the font **Source** directory.

3 Click **Convert**.

You should not edit the Destination field. Zebra fonts will be written to the Destination directory you specified in the Directory configuration. The destination files have the extension ZSU.

Note: Do not edit the ZSU file.

Editing the driver file and adding the fonts to be downloaded to the printer

To specify the font to be downloaded to the printer, you must edit the `Downloadable fonts` section in the driver file. For information on how to edit a driver file see the *Device driver tools* documentation.

Example 45 Specifying a font to be downloaded in the driver file

If you use the Book Antiqua Bold Italic font, `ANTQUABI.TTF` and you have the `ANTQUABI.ZST` in the fonts directory, you should add the following line in the `Downloadable fonts` section in the driver file:

```
Font "Book_Antiqua_Bold_Italic" ReadFont "antiquabi.ttf"  Select "antiquabi.zst"
```

Printing white text on black background

Avoid using colors in PageOUT when printing white text on a black background. Bright and dark contrasting colors may look white and black respectively, but may not give the expected result when printed out.

To print white text on black background

1. Create a rectangle with black fill color and black stroke color.
2. Enter the desired text on the rectangle and set text color to white. The text should not exceed the rectangle boundaries.

Barcodes

You configure barcodes in PageOUT. Supported barcodes are listed in *Table 1* on page 107.

Tips and caveats

- Barcodes, like text, can only be rotated 0, 90, 180, and 270 degrees. Any other angle will result in 0 degrees rotation in the printed output.
- Exact positioning of barcodes which are rotated any other angle than 0 degrees, is supported only for 1-dimensional barcodes.
- In the current PageOUT version, all features are not available for all barcode types. If an option is not available in the Barcode dialog box, the default value, as specified in the *ZPLII Programming Guide*, is used.
- For the Maxicode barcode, the encoding mode is determined by the data content. Only modes 2, 3, and 4 are supported. No structural append is available.
- For the UPC-E barcode, the length of the input data must be 11 or 12 digits with a pre-calculated checksum. The checksum must be in the same format as data for UPC-A. The system ID digit must be 0.

Supported barcodes

Table 1 Supported barcodes and their corresponding Zebra instructions.

Zebra instruction	Barcode
^B1	Code 11 (USD-8)
^B2	Interleaved 2 of 5
^B3	Code 39 (USD-3 & 3 of 9)
^B4	Code 49
^B7	PDF417
^B8	EAN-8
^B9	UPC-E
^BA	Code 93
^BB	CODABLOCK A, E, F
^BC	Code 128
^BD	UPS MaxiCode
^BE	EAN-13
^BF	Micro-PDF417

Zebra instruction	Barcode
^BI	Industrial 2 of 5
^BK	ANSI Codabar
^BL	LOGMARS
^BM	MSI
^BP	Plessey
^BQ	QR Code
^BU	UPC-A
^BX	Data Matrix
^BZ	PostNet

Graphics

The main difference between firmware versions X.9 and X.10 is in graphics support. Firmware X.9 supports fewer ZPLII commands than X.10. See *Table 2* on page 109, and the *ZPLII Programming Guide* for more information.

Non-supported graphical objects will not be included in the printed output.

Table 2 Differences in graphics support between firmware X.9 and X.10

Graphical object	Firmware X.10	Firmware X.9
Line	All line types are supported. For diagonal lines, the line width can differ depending on the slope angle.	Only vertical and horizontal lines are supported.
Rectangle	Supported. Target fill color is black (for input fill color with intensity 0-127 on a 0-255 scale) or white (for input fill color with intensity 128-255 on a 0-255 scale).	Supported. Target fill color is black (for input fill color with intensity 0-127 on a 0-255 scale) or white (for input fill color with intensity 128-255 on a 0-255 scale).
Rectangle with rounded corners	Supported. Target fill color is black (for input fill color with intensity 0-127 on a 0-255 scale) or white (for input fill color with intensity 128-255 on a 0-255 scale).	Not supported. Rounded corners are replaced by sharp corners in the printed output.
Circle and ellipse	Supported. Target fill color is black (for input fill color with intensity 0-127 on a 0-255 scale) or white (for input fill color with intensity 128-255 on a 0-255 scale).	Not supported.
Polygon/Polyline	Supported. Cannot be filled.	Not supported.
Path	Paths are always presented as polygons or polylines. Curves are presented as lines connected at the end points.	Not supported.

Enabling hexadecimal character values

If you want to use ZPL reserved characters (e.g. ^ and ~) in field data output to a Zebra label printer, you must use the corresponding hexadecimal character values.

To be able to use hexadecimal character values in field data (in an ^FD command), you must add the following keyword to the driver file (without parameters):

ENABLE_HEX

For example, to be able to use ~ (tilde) in output to a Zebra label printer, you must add the ENABLE_HEX keyword to the driver file, and use the hexadecimal value for ~ (<7e>) in the PageOUT Process. In the output to the printer, the field data is translated to the correct ZPL syntax. The table below shows the field data in PageOUT and the corresponding field data sent to the printer.

Field data in PageOUT	ZPL command
Tilde <7e> used for hex	^FH^FDTilde _7e used for hex^FS

Zebra Label Printer Settings

Driver settings	
Firmware version	The printer firmware version. Available options are X.10 and X.9 (X.9 means all firmware versions up to 9). To determine the version of the firmware in your printer, you can print out a Printer Configuration Label. See the printer documentation. The firmware version usually has the format `VXX.N.Z.` where `VXX` is the main version number (X), `N` is 9 or 10, and `Z` is a version extension.
Print mode	**Tear off** – The printed label is advanced so that it can be torn off manually. The backing is still attached to the label. **Peel off** – The backing is partly separated from the label. Printing stops until the label is removed. **Cutter** – The label is cut by the cutter mechanism. **Rewind** – The label and backing are rewound on an (optional) external rewind device. The next label is positioned under the printhead (no backfeed motion). **Applicator** – When used with an application device, the label move far enough forward to be removed by the applicator and applied to an item. **None** – Do not set the print mode.
Media type	**None** – Do not set the media type. **Direct thermal** – The media is heat sensitive and no ribbon is required. **Thermal transfer** – Black or color high carbon ribbon. The ink on the ribbon is bonded to the media.
Media tracking	**Non-continuous** – The media has some kind of physical characteristic, for example holes or perforations, that can be detected by the printer to separate the labels. **Continuous** – The media has no physical characteristic that can be used to separate the labels.

Driver settings	
Media feed at power up	The media feed action at start up. Available options are: • **Media feed** • **No media feed** • **Set media sensor calibration** – Feeds the media one label length, and recalibrates the media and ribbon sensors. • **Set label length** – Feeds one or more blank labels, depending on the label size. When using continuous media, the label length is defined by the page height, as defined in PageOUT.
Media feed at head close	The media feed action after closing print head. Available options are: • **Media feed** • **No media feed** • **Set media sensor calibration** – feeds the media one label length, and recalibrates the media and ribbon sensors. • **Set label length** – Feeds one or more blank labels, depending on the label size. When using continuous media, the label length is defined by the page height, as defined in PageOUT.
Label home X-axis	The label printing position along the x-axis. The default home position of a label is the upper-left corner (position `0,0`). Valid values are `0` - `300`.
Label home Y-axis	The label printing position along the y-axis. The default home position of a label is the upper-left corner (position `0,0`). Valid values are `0` - `300`.
Label top alignment	Moves the entire label up or down. The Label top is relative to the current Label home position. A positive value moves the format downwards, away from the top, while a negative value moves it upwards, towards the top. Valid values are `-64` - `64`. Default is `0`.
Tear off adjust alignment	Defines where the media is cut. The tear-off position is relative to the end of the printing on the label. Valid values are `-64` - `64`. Default is `0`.

Driver settings	
Media darkness	Adjusts the darkness relative to the darkness settings specified on the Printer Configuration Label. For example, if the value specified on the configuration label is `16`, and you specify Media darkness `-9`, the value will be `7`. Valid values are `-30` - `30`. Default is `0`. The maximum and minimum values cannot be surpassed. For example, if the value specified on the configuration label is `25` and you specify Media darkness `10`, the value will still be `30`.
Print rate	The media speed during printing. Valid values depend on the firmware version: Firmware X.9: `2`, `3`, `4`, `5`, `6`, and `8`. Firmware X.10: `2`, `3`, `4`, `5`, `6`, `8`, `9`, `10`, `11`, and `12`.
Slew rate	The slew speed (feeding a blank label). Valid values depend on the firmware version. See Print rate.
Print orientation	**Normal** – Prints the label with normal orientation. **Inverted** – Inverts the label 180 degrees, so that the label is printed upside down.
International Character Set	The character set used for printing. If you select a value other than `Auto`, the code page settings defined for the output connector will be ignored. If you select **None**, the international character set command in ZPLII output is suppressed. The label printer will use the last value that was saved. If you select **Auto**, the code page settings defined for the output connector are used to select the appropriate character set. **Note:** If you use international characters, you must configure the Zebra printer to use Unicode characters, see *Downloading TrueType and Unicode fonts to the Zebra printer* on page 103.
Image dithering method	The dithering method used for converting true color and palletized images to black and white images. Usually, **None** is used for black and white line art images, and one of the dithering methods are used for photographic images.

Driver settings	
Memory device for downloads	The device used for storing downloaded files, for example fonts and images. **R: (printer DRAM library)** – Always present and read/write access. **B: (optional memory card)** – A card or factory installed memory. Read/write access. **E: (flash memory)** – Use only with jobs using already loaded images and fonts. Read only access.
Pause and cut value (^PQ,p)	The number of labels to be printed before pausing for a cut. Valid values are `0` to `99999999`. The default value is `0`.
Copies	The number of labels to be printed.

Specifying fonts

The fonts available to the Process tools for page formatted output, and to the Overlay Editor, are the fonts included in the Windows `Fonts` directory on the machine where you configure the Project. This means you must make sure that all fonts you intend to use in your document designs, are included in the Windows `Fonts` directory before you try to include them in your documents. All used fonts must also be included as font entries in the driver files (`*.drs`) of the device drivers used by the StreamServer when delivering the documents. If you are using Type1 fonts, you must also import the font from the Windows `Fonts` directory to a resource set connected to the Message that uses the font.

Driver file font entries

A driver file can contain font entries, character width tables, and font mapping entries.

Example 46 *Readfont entry*

```
Font "Century" ReadFont "CENTURY.TTF"
SelectPrefix "Century" Codepage "Ansi"
```

Example 47 *Font entry with character width table*

```
// Century - regular, italic
WidthTable 1 Size 7200
7200  5400  5400  5400  5400  5400  5400  5400  5400  5400...
...
End
Font "Century" WidthTable 1
SelectPrefix "Century" CodePage "Ansi"
```

Example 48 *Font mapping entry*

```
mapfont "Times_New_Roman_bold_underline" "Arial_bold"
```

How the StreamServer handles fonts

The StreamServer uses two tables when it handles fonts:

- The font table that includes the available fonts.
- The mapfont table that includes font mappings.

When the StreamServer finds a font, it looks in the mapfont table first to see if the font is mapped to another font, and then it looks in the font table. If the font is found in the mapfont table, the StreamServer will use the mapped (new) font. If the font is found in both the mapfont and the font table, the StreamServer will still use the mapped font.

In previous StreamServe versions, the StreamServer looked in the font table first and then in the mapfont table. If the font was found in the font table, the StreamServer used that font even if the font was also included in the mapfont table (i.e. mapped to another font). This difference may cause errors if you run upgraded Projects with custom device drivers. For example, if the custom device driver includes a readfont entry for the font Century and also a mapfont entry that maps Century to Arial, the old Project version would use Century. The upgraded version of the Project will use the mapfont entry Arial instead.

Adding new font entries to driver files

If the driver file does not include entries for all fonts you are using, you must add the missing font entries to the driver file. See *Adding font entries to driver files* on page 117 for information on how to add font entries.

Making fonts available to the printer

All fonts included in the output from the StreamServer must be available to the printer (or equivalent) that receives the output. If there are fonts that are not installed on the printer, you must make them available to the printer. See *Embedding fonts and downloading soft fonts* on page 121.

Adding fonts to a resource set

If you move a Project from one machine to another, you cannot be sure that all used fonts are installed in the Windows `Fonts` directory of the new machine. In this case you can add all used fonts to the default resource set of the Project before you move it. The easiest way to do this is to export the Project, and check the option **Add exported fonts to Platform's default resource set** in the Export dialog. During the export, all used fonts are automatically added to the default resource set.

Adding font entries to driver files

All device drivers available to Design Center are stored in the device driver repository, and the font entries are included in a `*.drs` file attached to each driver. You cannot edit the default device drivers in the repository, but you can create a custom device driver by copying a default device driver. See the *Device Driver tools* documentation for more information.

Adding font entries to a custom *.drs file

You can add font and font mapping entries to any custom `*.drs` file in the device driver repository. When you export the Project, the new font and font mapping entries are included in the exported `*.drs` file.

Adding font entries to a font substitution table

You can also add font and font mapping entries to a font substitution table resource. When you configure the driver settings for the output connector, you add this font substitution table to the connector. When you export the Project, the new font and font mapping entries are appended to the exported `*.drs` file. If you use font substitution tables for several connectors with the same driver, only the font and font mapping entries from the last table are appended to the exported driver file.

Adding TTF font entries to driver files

You can use the Windows Driver Tool to create new TTF readfont and table width font entries.

To create font entries using the Windows Driver Tool

1. From the **Start** menu, select **<*StreamServe*>** > **Utilities** > **Windows Driver Tool**. The Windows Driver Tool opens.
2. From the **Printers** drop-down list, select any printer.
3. Specify the path to the driver file in **File name (drs-file)**.
4. Specify the path to the option file in **File name (opt-file)**.
5. Enter a **Driver name**.
6. Select **Create width tables during runtime** if you want to create readfonts.
7. Select the appropriate **Charset**.
8. Select the **Fonts** you want to create (you can multi select font entries) and click **Create**. You can now open the created `*.drs` file and copy the font entries.

True Type Collections (TTC)

You can use True Type Collections (TTC). TTC support is necessary for certain Japanese and non-Western typefaces. All information and instructions regarding TTF fonts in this document also apply to True Type Collections.

If the True Type Collection has an English name and a locale specific name (for example a Japanese name), you must update the driver file with separate entries for both of these names. For some languages, you must run the language version of Windows to be able to see the locale specific names.

Adding TTF font entries to PostScript, PDF, and raster driver files

This section applies to the `PDF`, `Postscript`, and raster drivers (for example the `BMP` driver). It does not apply to the `PDF (PCL convert)` driver.

To add font entries

1. Create the font entries using the Windows Driver Tool.
2. Open the dummy `*.drs` file and copy the font entries.
3. Paste the entries to the driver file or to the font substitution table.

To add font mapping entries

1. Open the driver file or font substitution table.
2. Add the font mapping entries. For example:

```
mapfont "Times_New_Roman_bold_underline" "Arial_bold"
```

Adding TTF font entries to PCL driver files

This section applies to PCL drivers except for the PCL 5 old version and PCL convert drivers.

Escape sequences

The PCL printer uses escape sequences to select fonts. Before you create font entries you must use the printer menu to print a list of all installed PCL fonts. This list includes the escape sequences for the fonts. The escape sequence for a proportional font can include the following:

```
<esc>(s1pv0s0b16602T
```

If you want to add the font as a readfont, you must replace `<esc>` with `<1B>` and divide the sequence into one `SelectPrefix` and one `SelectPrefix` section:

```
SelectPrefix "<1B>(s1p" SelectPostfix "v0s0b16602T"
```

Example 49 Readfont entry for the TTF font Century

```
Font "Arial" ReadFont "arial.ttf"
SelectPrefix "<1B>(s1p" SelectPostfix "v0s0b16602T"
Codepage "Ansi"
```

The escape sequence for a monospaced font can include the following:

```
<esc>(s0p16.67h15v0s0b3T
```

If you want to add the font as a width table font you must replace `<esc>` with `<1B>` and specify the size and CPI for each font size entry:

```
Size NN SelectPrefix "<1B>(s0p10h10v0s0b3T" CPI NN
```

Example 50 Width table font entries for the TTF font Courier

```
Font "Courier" WidthTable 8
Size 10 SelectPrefix "<1B>(s0p12h12v0s0b3T" CPI 12
Size 12 SelectPrefix "<1B>(s0p10h10v0s0b3T" CPI 10
```

To add font entries

1. Create the font entries using the Windows Driver Tool.
2. Open the dummy `*.drs` file and copy the font entries.
3. Paste the entries to the driver file or to the font substitution table. For example:

   ```
   Font "Arial" ReadFont "arial.ttf"
   SelectPrefix "Arial" Codepage "Ansi"
   ```

4. Replace the `SelectPrefix` entries with the modified escape sequences. For example:

   ```
   Font "Arial" ReadFont "arial.ttf"
   SelectPrefix "<1B>(s1p" SelectPostfix "v0s0b16602T"
   Codepage "Ansi"
   ```

To add font mapping entries

1. Open the driver file or font substitution table.
2. Add the font mapping entries. For example:

   ```
   mapfont "Times_New_Roman_bold_underline" "Arial_bold"
   ```

Adding Type 1 font entries to driver files

You can add Type 1 font entries to the following drivers:

- `PDF` (not `PDF (PCL convert)`)
- `Postscript`

You cannot use the Windows Driver Tool to create Type 1 font entries. This means you must add the entries manually to the driver file or to the font substitution table. Use the following syntax:

```
Font "STRS font name" ReadFont "fontfile.pfb"
SelectPrefix "Typeface name"
```

Where *STRS font name* is the font name in the exported *.dux, *fontfile* is the Type 1 font file, and *Typeface name* is the typeface name specified in the *.pfm font file.

Example 51 Minion Condensed Type 1 font entry

```
Font "Minion_Condensed" ReadFont "moc_____.pfb"
SelectPrefix "Minion Condensed"
```

To add font entries

1. Open the driver file or font substitution table.
2. Add the font entries.

To add font mapping entries

1. Open the driver file or font substitution table.
2. Add the font mapping entries. For example:

```
mapfont "Times_New_Roman_bold_underline" "Arial_bold"
```

Adding character encodings to Type1 font entries

The default character encoding for the fonts in the PDF driver is Windows code page 1252, and the default character encoding for the fonts in the Postscript driver is the default character set specified for the font.

To use other encodings outside the default character encoding for a font, you must add the new encoding to the font entry in the driver file. Character encodings are defined in separate encoding files (*.enc).

Adding an encoding file to a device driver

To make an encoding file available to the StreamServer, you must add the encoding file to the custom device driver. See *Adding device driver files* in the *Device driver tools* documentation for more information. When you export and deploy the Project, the encoding file is added to the drivers directory.

Adding an encoding to a font entry

You must update the readfont entry (either in the driver file or in a font substitution table) to include the EncodingFile keyword and the path to the encoding file.

Example 52 Adding an encoding to a font entry

```
Font "Minion_Condensed" ReadFont "moc_____.pfb"
SelectPrefix "Minion Condensed" EncodingFile
"drivers\iso8859_1.enc"
```

Embedding fonts and downloading soft fonts

If your output contains fonts that are not installed on the printer, you must make sure the missing fonts are sent to the printer.

Embedding fonts

You can embed the missing fonts in the output sent to the printer. See *Embedding fonts in PDF documents* on page 121 and *Embedding fonts in output to PCL printers* on page 121.

Downloading soft fonts

Embedding fonts in the output increases the size of the output files. If you want to keep down the size of the output sent to PCL printers, you can download the missing fonts as softfonts to the printer. See *Downloading soft fonts to PCL printers* on page 122.

Embedding fonts in PDF documents

You can embed TTF, TTC, and Type 1 readfonts in PDF documents. Note that this is applicable only to the `PDF` driver, and not to the `PDF (PCL convert)` driver. To embed a font, you add the `FontEmbed` keyword to the entries for the fonts you want to embed.

Example 53 *Embedding a font*

Original font entry:

```
Font "Arial" ReadFont "arial.ttf"
SelectPrefix "Arial" Codepage "Ansi"
```

Modified font entry:

```
Font "Arial" ReadFont "arial.ttf"
SelectPrefix "Arial" Codepage "Ansi" FontEmbed
```

To embed a readfont

1. Open the driver file or font mapping table that includes the entries for the fonts you want to embed.
2. Edit the entries for the fonts you want to embed.

Embedding fonts in output to PCL printers

You can embed TTF readfonts in the output sent to PCL 5 compatible printers that support True Type based soft fonts. To embed a font, you add the `FontEmbed` keyword and the escape sequence `SelectPrefix "<1B>(s" SelectPostfix "V"` to the entries for the fonts you want to embed.

Example 54 Embedding a font

Original font entry:

```
Font "Arial" ReadFont "arial.ttf"
SelectPrefix "<1B>(s1p" SelectPostfix "v0s0b16602T"
Codepage "Ansi"
```

Modified font entry:

```
Font "Arial" ReadFont "arial.ttf" FontEmbed
SelectPrefix "<1B>(s" SelectPostfix "V" Codepage "Ansi"
```

To embed a readfont

1. Open the driver file or font mapping table that includes the entries for the fonts you want to embed.
2. Edit the entries for the fonts you want to embed.

Downloading soft fonts to PCL printers

You can download missing fonts as soft fonts to PCL printers if you want to keep down the size of the output sent to the printer. The downloaded soft fonts are available to the printer until it is restarted. If the printer is restarted, you must download the soft fonts again.

TTF soft font files

You can download TTF fonts as soft fonts to the printer. Please contact your StreamServe supplier for information on how to acquire soft fonts. The soft fonts must have a format supported by the printer, and include the same character set as specified for the output. For example, if you use Russian version TrueType fonts in the Project, the corresponding soft fonts must include the Russian version.

Type 1 soft font files

You can download Type 1 fonts as soft fonts to the printer. To create a Type 1 soft font file, you must convert the binary `*.pfb` font file to an ASCII `*.pfa` font file. Then you add the following section at the beginning of the `*.pfa` file:

```
%!load the font in VM
serverdict begin
0
exitserver
```

How to convert `*.pfb` files to `*.pfa` files is not described in this document.

To download soft fonts

1. In Design Center, activate the generic Platform layer, and double-click the output connector that delivers the output to the PCL printer. The Output Connector Settings dialog opens.
2. Click the **General** icon and select **Enable download**.
3. In **Download File**, specify the soft font file to download.

4 Export the Project, start the StreamServer, and send output to the PCL printer. The soft font is downloaded to the PCL printer.

When you have downloaded the soft fonts, you must use the printer menu to print a list of all installed PCL fonts. This list includes the escape sequences for the downloaded soft fonts. Then you must add the new soft font entries to the driver file. See *Adding TTF font entries to PCL driver files* on page 118.

Enabling soft fonts for the PDF (PCL convert) driver

If you are using the PDF (PCL convert) driver, you must add the following custom argument at the beginning of the startup argument file:

```
-pcl2pdfarg -ru:1
```

RFID

Radio Frequency Identification (RFID) tags are used for automatic identification of individual items. RFID inlays are configured in the PageOUT tool and the following StreamServe drivers can be used for sending RFID configuration information to label printers:

- Intermec DP (203dpi)
- Intermec DP (300dpi)
- Zebra ZPLII (150dpi, 200dpi, 300dpi and 600dpi)
- Printronix IGP/PGL

Prerequisite

To configure RFID inlays you must have:

- Detailed knowledge in the printing environment and the hardware configuration.
- Basic knowledge in RFID, such as, air protocols and tag data formats (EPC encoding).

References

- EPC Class 1 Generation 2 and EPC tag data standards:
  - http://www.epcglobalna.org/StandardsDevelopment/EPCglobalStandards/tabid/185/Default.aspx
- Zebra references:
  - http://www.zebra.com/
  - http://www.zebra.com/id/zebra/na/en/index/products/printers/rfid.html
- TEC references:
  - http://www.toshibatecusa.com/
  - http://www.toshibatecusa.com/Products/RFIDProducts/tabid/60/Default.aspx
- Printronix references:
  - http://www.printronix.com/
  - http://www.printronix.com/rfid-printers.aspx
- Intermec references:
  - http://www.intermec.com/
  - http://www.intermec.com/products/rfid/index.aspx

Known limitations

Printronix:

- No support for ISO18000-6b.
- No support of ASCII-96, ASCII -64 and NUM-96.
- No support to lock EPC Class 1 inlays, only kill password is supported.

Intermec:

- Limited support of NUM format. Maximum 4 bytes, 10 digits.
- NUM format is not supported for EPC Class 1 air protocols.

Zebra:

- No support for NUM format.

Supported air protocols and printer models

The following air protocols are supported:

- EPC Class 1 Version 1
- EPC Class 1 Generation 2
- ISO18000-6b

The printer models listed in the table below are supported.

Printer model	Air protocol
Intermec EasyCoder PM4i f/w IPL and f/w FingerPrint/Direct protocol	EPC Class 1 Version 1 EPC Class 1 Generation 2 ISO18000-6b
Zebra R110Xi (SP994X)	EPC Class 1 Version 1 EPC Class 2 Generation 2 ISO18000-6b
Zebra R4MPlus f/w: ZPL II (R60.13.X)	
Printronix SL5000r MP, SL5000r MP2 f/w PG	EPC Class 1 Version 1 EPC Class 1 Generation 2

RFID settings

The following settings are available for the drivers supporting RFID. You configure the settings in the Platform configuration.

Note: The settings are not applicable for Printronix.

RFID settings	
RFID antenna offset	The distance the label is moved to align the inlay over the coupler (the printer antenna) before programming the chip. Depending on the printer model, specify antenna offset in dots or millimeters.
RFID number of retries in dots	Can be set to `0-10`. **Intermec printers** If set to 1 or greater and a read or write command fails, the printer tries to read or write the next label. If this also fails the procedure is repeated until a successful read or write command is committed, or the specified number of retries is reached. The string VOID, or any other selected string, is printed on all rejected labels. **Zebra printers** The number of read or write retries before a transponder is declared defective. When the specified number of retries is reached, the string VOID is printed on the label and the transponder is ejected.
Print mode (^MM)	Only applicable for Zebra. **RFID** – Activates the RFID functionality in the printer.